PROJECTE FINAL DE MASTER

Implementation of a Firefox Extension that Measures User Privacy Risk in Web Search

***Estudis**: Máster en Ingeniería Telemática*

***Autor**:* José Antonio Estrada Jiménez

***Director**/a:* Jordi Forné Muñoz, PhD

***Codirector**/a:* Javier Parra Arnau, PhD (c)

***Any**:* 2013

Index

Index of figures

Agradecimientos

Este trabajo, como todo lo que hago, está dedicado a mis padres Antonio y Laura, y a mi abuela Celia, en honor a su infinito amor y apoyo.

A Anita, mi compañera de vida, por su amor incondicional y sacrificio, por la paciencia que me ha tenido estos dos años.

Al profesor Jordi Forné y a Javier Parra, por la guía invaluable que ha hecho posible este trabajo.

A mis hermanos Juan Carlos, María Cristina y Luis Alberto por estar cerca, aún a la distancia; y a todos mis amigos cuya existencia hace que la cosecha de estos frutos sea aún más dulce.

A la Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) de mi país, Ecuador, que es el brazo ejecutor de la política estatal que auspició con una beca completa mis estudios de Maestría.

Resum del Projecte

Actualment, la monitorització dels usuaris a Internet és permanent, i la informació obtinguda en aquest procés és d'enorme interès per a grans companyies de publicitat i especialment per governs. Entre la informació d'usuari que agents externs recopilen està fonamentalment: les consultes de cerca, els clics, la informació de pàgines visitades i fins el temps de visita de llocs web. Aquestes dades són processades amb la finalitat d'obtenir perfils d'usuari en base als quals ara una gran quantitat de sistemes web personalitzen els serveis que ofereixen.

A més, la gran quantitat de dades susceptibles de recopilar pels sistemes d'informació personalitzats representa un greu risc per a la privacitat de l'usuari a Internet. Potser encara més crític és que molts usuaris no són conscients d'aquest risc, ja que aquest no és tan manifest com en el món físic. Aquest risc s'agreuja quan poques empreses poden concentrar gran part de les dades d'usuari ja que els seus serveis són molt populars i arriben a ser imprescindibles per a la interacció amb Internet, com és el cas dels motors de cerca. Tot i que els usuaris no revelessin informació estrictament personal, com adreces, ubicacions o noms, la tecnologia actual permet inferir gran part d'aquesta informació a partir de cada interacció de l'usuari amb Internet.

En aquest treball presentem una aproximació per al desenvolupament d'una extensió del navegador Firefox que estima el risc de privacitat del perfil d'una persona, que, pels seus hàbits de navegació, està exposat a mecanismes de profiling a Internet. El nivell de risc es mostra, de manera entenedora i accessible a la interfície gràfica del navegador i es calcula tenint en compte diferents models d'adversari.

El mecanisme de mesurament de privacitat que integrem al navegador utilitza mètriques adequadament justificades i basades en conceptes de teoria de la informació.

Per facilitat d'ús, aquesta eina disposa d'una barra informativa de privacitat directament integrada en el navegador, i per tant és permanentment visible mentre el navegador és utilitzat. Addicionalment mitjançant finestres es desplega informació més detallada respecte dels nivells de privacitat de l'usuari, tant individualment com en comparació amb altres models de perfils de població.

Paraules Clau: perfil d'usuari, mètriques de privacitat, entropia de Shannon, divergència de Kullback-Leibler, extensió de navegador, add-on de navegador.

Resumen del Proyecto

Actualmente, la monitorización de los usuarios en Internet es permanente, y la información obtenida en este proceso es de enorme interés para grandes compañías de publicidad y especialmente para gobiernos. Entre la información de usuario que agentes externos recopilan está fundamentalmente: las consultas de búsqueda, los clics, la información de páginas visitadas y hasta el tiempo de visita de sitios web. Estos datos son procesados con el fin de obtener perfiles de usuario en base a los cuales ahora una gran cantidad de sistemas web personalizan los servicios que ofrecen.

Además, la gran cantidad de datos susceptibles de recopilarse por los sistemas de información personalizados representa un grave riesgo para la privacidad del usuario en Internet. Quizá aún más crítico es que muchos usuarios no son conscientes de este riesgo, ya que éste no es tan manifiesto como en el mundo físico. Este riesgo se agrava cuando pocas empresas pueden concentrar gran parte de los datos de usuario ya que sus servicios son muy populares y llegan a ser imprescindibles para la interacción con Internet, como es el caso de los motores de búsqueda. Aunque los usuarios no revelaran información estrictamente personal, como direcciones, ubicaciones o nombres, la tecnología actual permite inferir gran parte de esa información a partir de cada interacción del usuario con Internet.

En este trabajo presentamos una aproximación para el desarrollo de una extensión del navegador Firefox que estima el riesgo de privacidad del perfil de un usuario, quien, por sus hábitos de navegación, está expuesto a mecanismos de *profiling* en Internet. El nivel de riesgo se muestra, de manera comprensible y accesible en la interfaz gráfica del navegador y se calcula tomando en cuenta diferentes modelos de adversario.

El mecanismo de medición de privacidad que integramos en el navegador utiliza métricas adecuadamente justificadas y basadas en conceptos de teoría de la información.

Para facilidad de uso, esta herramienta dispone de una barra informativa de privacidad directamente integrada en el navegador, y por lo tanto es permanentemente visible mientras el navegador es utilizado. Adicionalmente mediante ventanas se despliega información más detallada respecto de los niveles de privacidad del usuario, tanto individualmente como en comparación con otros modelos de perfiles de población.

***Palabras Clave*:** perfil de usuario, métricas de privacidad, entropía de Shannon, divergencia de Kullback-Leibler, extensión de navegador, *add-on de navegador*.

Abstract

At the present time, user activities on Internet are permanently monitored, and the information obtained from this process is pretty useful for big advertising companies and especially for governments. The user information that external agents collect could be: search queries, clicks, text from visited web sites and even the time that users spend on browsing some sites. These data are processed in order to build user profiles that are later used to customize services.

Additionally, this large amount of data susceptible of being collected (by personalized information systems) represents a serious risk for the users' privacy on Internet. Perhaps, even more critical is the fact that many users are not aware of this risk, since it is not as evident as it is in the physical world. The risk gets worse when a few companies are capable of collecting much of the user data because their services are so popular to the point where they become vital for users interaction with the Internet, as is the case with search engines. Even when users do not reveal strictly personal information such as addresses, locations or names, current technology allows to infer much of this information from each user interaction with Internet.

We present in this work a first effort towards the development of an extension for the Firefox web browser that estimates the privacy risk of the privacy of a user who, due to its navigation habits, is exposed to profiling mechanisms on Internet. The risk level is shown in a comprehensible and accessible fashion in the browser graphical interface and is calculated based on different adversary models.

The privacy measuring mechanism that we integrate in the browser uses metrics that were appropriately justified and that rely on information theoretic concepts.

For ease of use, this tool, that we called *PrivMeter*, provides a privacy informative bar that is directly integrated in the browser's main graphical interface, and therefore it is permanently visible while the web browser is used. Furthermore, by means of windows *PrivMeter* displays more detailed information about the privacy levels of the user, both individually considered and also comparing it with other population privacy data.

Keywords: user profile, privacy metrics, Shannon's entropy, Kullback-Leibler's divergence, browser extension, browser add-on.

1. Introduction

1.1 Motivation

Nowadays, a vast amount of information is generated on the Internet, from mobile devices, applications and websites to the point where incredible amount of data is collected by different systems globally. Also, due to the extreme ease of generating content, regular users have also become responsible for generating much of these data. Several companies have realized the great potential of analyzing this information towards the achievement of advertising objectives. This business is so successful that these companies are even offering “free” services with the objective of getting user data.

The evident advances in data analysis and the increasing capabilities of collecting personal information have helped to profile and classify users more easily than ever. Also, personal information retrieval and profiling are common practices of systems that process all the user content on Internet in order to understand preferences and habits, to then be capable of providing a completely personalized service.

This means that the simplest web interactions are gradually detected and measured to build detailed collections of user behavioural data. The advent of social networking and the deep degree of data *linkability* on Internet have also facilitated the tracking of user’s trends and interests.

There is no doubt that creating user profiles based on navigation patterns have profoundly changed the user experience on Internet, especially with the objective of offering personalized recommendation of content and advertising.

But accurate recommendation of content requires mechanisms to track users on the Web to get as many user data as possible. The price of a more dynamic and richer access to information is, therefore, very high: the user privacy. Traces left by a user, although scattered, perturbed or combined with others from different sources, could reveal potentially sensitive information associated with personal preferences, as it is described in [1] and [2].

The information susceptible to analysis includes the content of visited web sites, consumed time, number of clicks, queries to search engines, data gathered from web forms, cookies, and even the footprints left by the web browser each time it is used to surf the Web [3].

There are multiple agents on Internet to which users yield their information to obtain a service. The user environment on Internet is, therefore, full of a wide range of potential attackers and it is not clear what is being done with all of the collected

data about users. These adversaries could be: search engines, recommendation systems, social networks, tagging systems, etc. The Internet Service Providers (ISPs) are themselves involved in this process of analyzing and profiling data, because they have complete access to all the information about user activity and, in some cases, this information is commercialized with advertisement companies without consideration on the owners of the data [4].

There is, in fact, so much pressure on the companies that manage personal information (see [5] and [6]) to apply privacy policies to protect sensitive data. However, external pressure (e. g. from governments) to reveal these digital traces seems to be greater.

Furthermore, privacy policies implemented by web service providers are so deficiently communicated that users barely read or understand them. So, people accept agreements so rapidly and without meditation, with the solely objective of immediately using some “free” service.

All these phenomena show that there is a generalized lack of awareness of the risks to which Internet users are exposed and the consequent infringement of their human right to privacy.

These economic and political interests over the information on the Web, and the inadequate practices related to user privacy do not seem to change. However, the user’s behavior regarding his information can be modified, if the weaknesses of this conduct can be made evident [28].

The fact is that there are not tools to inform the user about their privacy level. There are some tools that implement information obfuscation or blocking mechanisms, but no one capable, at least, of establishing its effectiveness. The problem is complex, because the privacy level may depend on the user’s environment (e.g. the adversaries) and perception, so the protection mechanisms may even depend on the user individual interests and considerations.

It is essential to measure the privacy risk level in order to accordingly apply protection mechanisms adapted to the user needs, especially when the huge success of targeted advertising encourage all the content providers to apply advanced techniques to elaborate profiles and model users behavior.

This work describes the design and implementation of an extension for the Mozilla Firefox browser, which we called *PrivMeter,* to measure the privacy of a user. By reusing some modules of the Adnostic [7] extension and by processing the user’s information collected in the browser, this extension shows multiple privacy measures. The knowledge and interpretation of these measurements could help a user to making a decision about their activity on the Web and allowing them to evaluate privacy protection technologies.

The privacy risk levels and related measurements are made by using metrics that are justified with concepts from information theory.

1.2 Objectives

The main objective of this work is to implement a tool capable of measuring and displaying the user's privacy while browsing on Internet. Essentially, the information exchanged between the user and web services will be used to obtain a user profile based on which the privacy measurement will be made.

Other specific objectives of this work are the followings:

- Take advantage of already available tools for capturing user information in the browser.
- Locally process user information in the browser to build a user's profile.
- Measure the privacy risk of the user based on his profile, with search engines and ISPs as adversaries.
- Use justified metrics for accomplishing the objective of privacy measurement.
- Implement a graphic interface to clearly show the results of the privacy measurement.
- Give a solid base of decision to the user when implementing a privacy protection strategy.

1.3 Organization of the memory

This work has been organized in the following way. Chapter 2 explores some of the available tools and mechanisms oriented to the user privacy protection, especially highlighting the lack of user tools to measure privacy. Chapter 3 describes the metrics used to determine the privacy level of the user and the adversary models based on which these metrics haven been analyzed. A clear justification is exposed for the usage of these metrics. Chapter 4 addresses the work of implementation of a Firefox extension capable of establishing the level of user privacy risk by analyzing the information the user delivers to Internet. Finally, in Chapter 5, some concluding ideas and recommendations of future work are mentioned.

2. Measurement and protection of user privacy on the Web

Currently, there are some tools trying to protect the user privacy on the Internet, essentially by blocking some browsing functions that commonly facilitate the leaking of personal information. Sadly, due to the enormous dependence of several Internet services on these tracking practices, a big number of web sites will not work correctly if some of these “features” are disabled.

As it was mentioned in the Introduction, apart from a few efforts to heuristically shield the user privacy when browsing the Web, there are no privacy measurement tools in the digital landscape. This results in an environment where common users are not aware of the risks that they are facing. Consequently, not having enough information makes it impossible for them to implement a successful strategy to protect their data.

2.1 Importance of the web browser for privacy measuring

Web browsers are the applications used by regular users to access Internet. This intermediary between the user and web sites translates user’s interactions into requests that web servers can understand. Working as local proxies of the user’s requests, web browsers are capable of “seeing” all the user’s information handed over the Web, so several approaches to protect the user’s privacy are implemented as browser extensions or add-ons. These extensions use the information collected from the user to locally analyze it and block risky connections or unnecessary executions of code.

In terms of security, offering locally implemented privacy protection mechanisms is much more appreciated, because disclosing data to third parties will increase the risk we are trying to manage.

Being user side agents, the role of the web browser and its extensions can be very active when measuring or guarding privacy.

2.2 Privacy risks on the Web

The user’s privacy on the Internet depends on the amount and quality of information that users “share”. Some common online activities through which these information is collected are the followings [8].

- **Signing up for Internet service**. When connecting to Internet, users sign a contract with an ISP, who provides the network access to the service. ISPs know the IP address assigned to their users, and are able to identify their customers by means of this parameter. The IP address is also an identifying factor when visiting web sites because it is visible along the whole path a packet takes to reach its destination.

- **Browsing on Internet.** Browsing involves the interaction of users with some very popular services on Internet by means of clicks and web form filling.

  Search engines are now part of the most popular services to which users daily send millions of queries to reach specific content. These queries are critical identifying data that accurately reveal topics on which users are very interested. But still more sensitive information is disclosed, since people search everything on Internet: health conditions, physical addresses, travel details and names of people of their interest.

  Cookies are data structures, which are stored in the web browser to help web services to easily identify the connections made by the same user. This allows a further customization of content, according to user preferences discovered during the previous interactions. The so called third-party cookies, however, communicate this information (i.e. interests associated to an electronic identity) to several online marketers who track the users to deliver ads.

  Blocking these cookies is relatively easy when using integrated facilities in the browser. That is why a new kind of cookie called Flash cookie has appeared in the landscape. This cookie cannot be immediately deleted because it is more resistant than the common ones.

- **Fingerprinting.** The browser's fingerprint, consisting on all the information collected from the browsing application, is also commonly used as an identifier instead of cookies. This fingerprint covers a complete detail of the software characteristics of the browser: versions, patches, operating system, add-ons, physical resources, etc. Unlike what we may think, this vector of information could contain enough data to identify an individual inside a region (as it is stated and showed in [3]).

- **Using mobile applications.** Given that much of the Internet traffic has started to be originated from mobile devices equipped with many sensors, the information "delivered" by the user to the network could be even more sensitive since it involves data about location, agenda, calls logs, contacts and more.

  Depending on the permissions granted to mobile applications, all this information might be visible to third parties, with the corresponding danger that this implies.

- **Using e-mail.** There is nothing new in mentioning that web mail services such as Gmail or Yahoo Mail (and probably others) scan the content of messages to then personalize ads according to this information. Additionally, on sending an e-mail (commonly using plain text) the user trusts his ISP, the e-mail provider and the recipient. All of these agents are able to read the user's message and, consequently, may exploit his privacy.

In general all these activities, in the same manner as instant messaging, social networking, blogging and online banking services usage, involve a risk of compromising user privacy due to the fact that all of them are based on the confidence that the user must have on third parties that store his personal information.

A number of potential adversaries are present on the Internet equivalent to the number of available services, as it is shown in the previous paragraphs. The most powerful ones, in terms of the volume of information that they are able to collect, are, with no doubt, search engines, social networks and ISPs.

As if this were not enough, the pressure on these companies (e.g. Google) is increasing. Governments and federal agencies of investigation periodically asks them for user's search and activity logs in a periodic basis, so the resistance against the private information disclosure keeps being threatened, especially when anonymizing such volumes of data is very hard.

2.3 Available privacy protection tools

Most privacy protection tools available for Internet navigation are designed to block some functionalities of dynamic web sites that enhance the user experience undermining their privacy. These mechanisms are generally based on heuristics and do not measure the user privacy risk nor evaluate the level of protection being offered, so it is uncertain whether these practices are really being effective. Moreover, blocking these facilities will, most of the time, prevent users from accessing to the mentioned web sites.

2.3.1 TrackMeNot

TrackMeNot (TMN) [9] is widely known among the privacy research groups in the academic world. It is a privacy protection tool implemented as a Firefox and Chrome extension that obfuscates queries sent by the user to some of the most popular search engines. In order to achieve this objective, TMN dynamically generates fake queries and mixes them up with real user queries in attempting to confuse potential adversaries such as the ISP or the search engines.

2.3.2 REPRIV

REPRIV [10] is proposed as a whole system to be integrated with the browser to offer an enhanced retrieving of personalized content and a mechanism to manage the information that the user delivers to third parties. As it is expected, REPRIV employs browsing information to derive the user's preferences. Third parties, providing personalized content, receive these preferences and model the user's content according to the corresponding user's profile.

Communication interfaces are also proposed for the websites to be able to process this information sent by REPRIV and by using HTTP protocol.

It definitely offers a considerable improvement in the quality of delivered content, thanks to the great detail of user's information obtained from the browser. This system apparently provides a complete control over security policies to manage the shared information. But, dealing with such policies would seriously affect the usability of the system.

2.3.3 Adnostic

Adnostic [7] is another extension developed for Mozilla Firefox that implements an architecture to display personalized advertising without compromising the user's privacy. The ads to be shown to the user are chosen in the browser, according to a locally estimated personal profile. This profile is constructed by processing the user's queries and the content of web pages he has visited. Then, this information is classified by means of natural language processing techniques inside the browser. The ads, which are part of a previously downloaded set, are displayed according to the user's interests.

2.3.4 Google Sharing

Google Sharing [11] is a tool that provides a privacy protection mechanism by means of avoiding the user tracking made by Google. This Firefox add-on works by redirecting user's requests to an external proxy where a group of identities associated with cookies are managed. These cookies replace the ones included in original requests, masking the user's identity, and are then forwarded to Google with the original request. Even when they allow users to send encrypted requests, the user's privacy can still be compromised if collusion exists between the proxy server and Google.

2.3.5 Private browsing

It is a privacy protection option offered by many of the most used web browsers. It allows users to browse on the Web without locally saving any information about what sites and pages they have visited. The storage of some other information such as

videos, images and cookies, is also disabled. This significantly complicates the access to several sites in Internet. That is why people who use this tool only do it during very short intervals of time.

The protection level is limited to the local scope because, since externally, there are other mechanisms to identify and classify the user's profile.

2.3.6 Blocking browser facilities

Deactivation of some characteristics in the browser is a common strategy implemented by some plug-ins such as NoScript [12], AdBlockPlus [13] and DoNotTrackMe [14]. All of them try to prevent the leaking of information that could be used to identify a user.

None of the mentioned mechanisms and tools evaluate the user's privacy level. Advertising systems and social networks are the only potential adversaries considered. Internet Service Providers, for example, even when they are the entities that have the greatest volume of information about users, they are not taken into account as attackers.

In [15], [16], [17] and [18], some theoretical mechanisms are studied that could be used to protect the user's privacy in environments where he sends queries or tags. It also considers the cost of these strategies which is reflected in the loss of data utility, the loss of service functionality or the additional required resources. Among these mechanisms, the query forgery or tag suppression are included in order to show a distorted version of the user's profile that the attacker cannot exploit. Optimization of these mechanisms and its impact are also analyzed.

3. Scenario and privacy metrics

This chapter illustrates the environment considered for the quantification of privacy level. This environment basically includes the user and adversary models used to establish the privacy metrics.

3.1 Motivating Scenario

In this project, we use a partial scenario from the one considered in [19]; where the mathematical analysis to quantify privacy and to model adversaries is still valid.

Let's imagine a common user, such as the millions of users daily accessing the Web to search about a subject of his particular interest. He uses a search engine (mostly Google) to get the URLs related to the content he is looking for. In fact, even knowing the URL of the web site he wishes to visit, this user prefers sending a short query to Google to get the URL, because it is faster for him. For regular users, search engines have become the application gateways to Internet services because they tremendously facilitate the navigation on the Web. The search results have such quality that are disambiguated according to user's interests, interests that have been learned from previous queries. The level of personalization achieved in such services is such that it appears that they know more about the user than the user themselves.

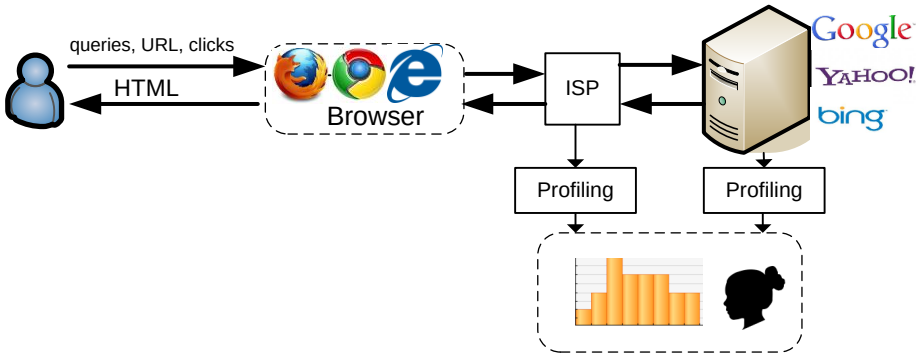

Fig. 1. Scenario where privacy metrics are implemented in this project, according to the mathematical analysis done in [19].

There is in this scenario an evident concern about the user's privacy, since there is a high probability that the users were not interested in revealing publically some details about their life.

Thus, the scenario is structured by search engines and Internet Service Providers as potential attackers. Search engines can visualize all the query strings sent by the user and the URLs he has clicked, and the ISPs are capable of accessing to all the information generated by the user while browsing on Internet. But not only that, these

monitoring agents are continuously profiling the owners of these data with commercializing purposes. As it is showed in Fig. 1, as Internet users, we are at the mercy of content gateways that apparently know us better than we do.

3.2 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 in which we are interested, so the words that are part of these queries strongly represent our 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 on each topic, depending on the number of queries, tags or clicks sent or created by the user with relation to such topics.

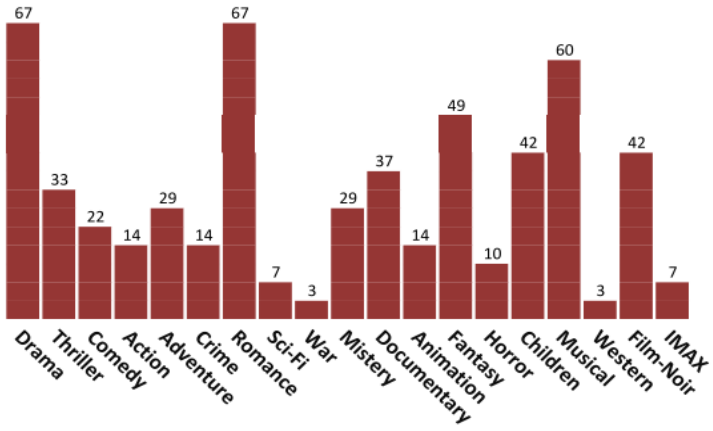

Fig. 2. Histogram modeling a user profile.

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 as it is shown in the Fig. 2.

3.3 Adversary models

The adversary model defines the properties of the attacker. An attacker is considered as any entity capable of having access to user's information in order to obtain his profile.

Defining and describing the adversary is very important since the privacy is quantified according to this entity. Depending on the adversary, a protection strategy could be more or less effective. This context, when doing a security analysis, is critical because all threat scenarios cannot be covered, so depending on the hypothesis that are done about the environment (e.g. the adversary), the used metrics may significantly vary.

According to the user's model previously defined, a user could spoil a profiling strategy by provoking changes in his histogram of interests (e.g. artificially modifying his normal browsing behavior). The attacker would not be able to obtain valuable information, since the user's profile would have been perturbed. This new version of the profile, from the user's perspective, is known as its apparent profile.

The adversary model can be defined by the attackers' capabilities and, in this sense, two objectives are considered:

- **Identification.** In this case, an attacker is attempting to identify a user in the sense of distinguishing this user from the rest of the population, by detecting deviations between user's interests with respect to the average profile of the population.
- **Classification.** An attacker tries to classify a user by comparing the user's profile with the representative profile of a group, in order to predict the group in which this user could be catalogued.

3.4 Privacy metrics

As justified in [19], Shannon's Entropy and Kullback-Leibler (KL) divergence are used as privacy measuring parameters. The interpretations of these parameters will essentially depend on the hypothesis made, with respect to the adversary model.

Another more general metric, not limited to profiles' privacy, is the one proposed in [20]. In this work, the authors propose measuring privacy as the estimation error of an adversary, and interprets, by means of information and Bayesian decision theory, other metrics of 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

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

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

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

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$_2$ *n,* being *n* the cardinality of the finite alphabet over which *D* (*p* || *u*) is calculated, for a uniform distribution *u*:

$$D(p \,||\, u) = \log n - H(p)\ .$$

Specifically, according to the analysis made in [19], 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 the user's profile, and $\bar{t}$ the distribution of population's profile, the interpretations of Shannon's entropy and KL divergence as privacy metrics are shown in Fig. 3. These ideas are detailed, with respect to the attacker's objective, in the following sections.

3.4.1 Metrics against identification

If the goal of the attacker is to identify the user, in the sense mentioned above, Janes' rationale about entropy maximization methods allows the justification of both the divergence and the entropy as measures of privacy.

The entropy of the user's apparent profile, that is, the profile observed by the attacker, is justified in [19] 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 [19] also justify the entropy as a measure of privacy. In brief, the higher the entropy of a profile, higher is its probability, and therefore greater is the number of users behaving according to this profile, which means that it is more private.

Furthermore, as it is observable in the first branch of Fig. 3, 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.

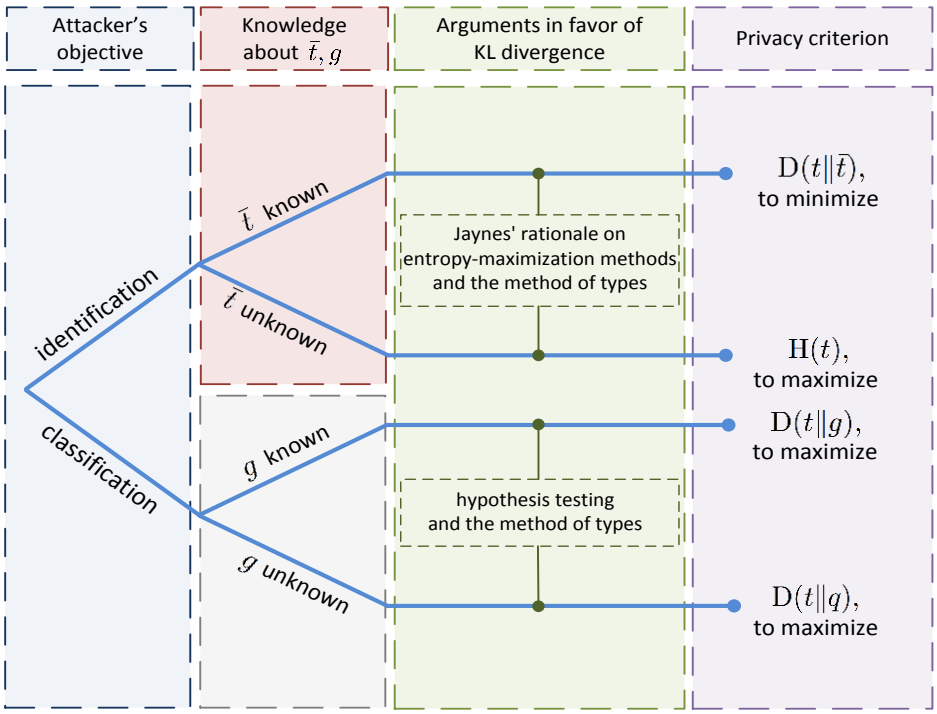

Fig. 3. Summary of the interpretations of Shannon's entropy and KL divergence as metrics of privacy, taken from [19].

3.4.2 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 [19], from hypothesis testing and the method of types. As shown in Fig. 3, in the second branch, if the profile of group *g* is unknown from the user side, the option is maximizing the divergence between the real profile *q* and the observed (apparent) profile *t*, in order to avoid being classified according to the original profile.

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.

4. Implementation of a Firefox extension for measuring privacy

As it was mentioned in Chapter 1, the purpose of this work is the implementation of a tool that measures the user's privacy in the browser, in order to show him his privacy risk level. Honoring its functionalities, we called this tool as *PrivMeter.*

The measurement of privacy, according to the metrics described in the last chapter, 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 Internet, as stated in Chapter 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 [21].

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

With the purpose of delimiting the scope of this project, this section defines the main premises for the development of the browser extension. Fig. 4 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. 4: the search engine service and the Internet Service Provider. 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.

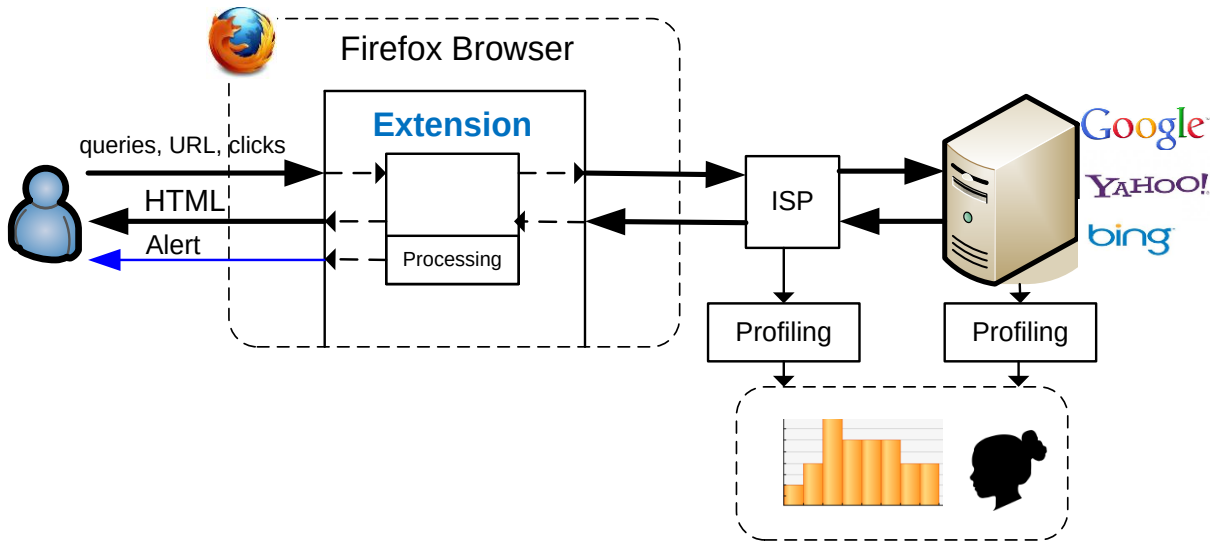

Fig. 4. Scheme of a browser and its extension as intermediaries between the user and Internet services (search engines and ISPs) showing the inherent risk of profiling.

The information obtained from the user patterns of browsing allow the attackers to build a detailed user profile. But, obtaining this profile is also crucial for the user in order to realize the risk he is facing when browsing on 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. 4) 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 the chapter 3, reference profiles must be available so that the discrepancy between these profiles and the user profile can be calculated.

The privacy measurements will be shown to the user as levels of risk, numerically expressed, and by means of colored graphs, in order to facilitate the comprehension of the different security parameters.

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. 5 illustrates this structure where the executed processes and the results are shown for each functional module.

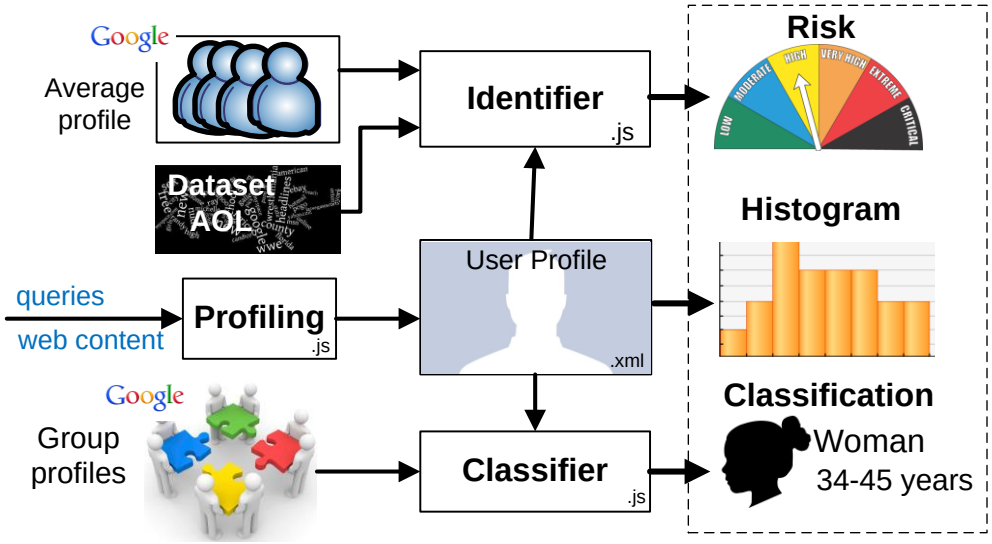

Fig. 5. Architecture for privacy level calculation.

4.2.1 Web Browser

As illustrated in Fig. 4, 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.

Let's remember that 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. 4, 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).

4.2.2 Profiler

Establishing the user's profile, consists on 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, in the same way as explained before in section 3.2 .

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. 6 illustrates the architecture of this profiling process.

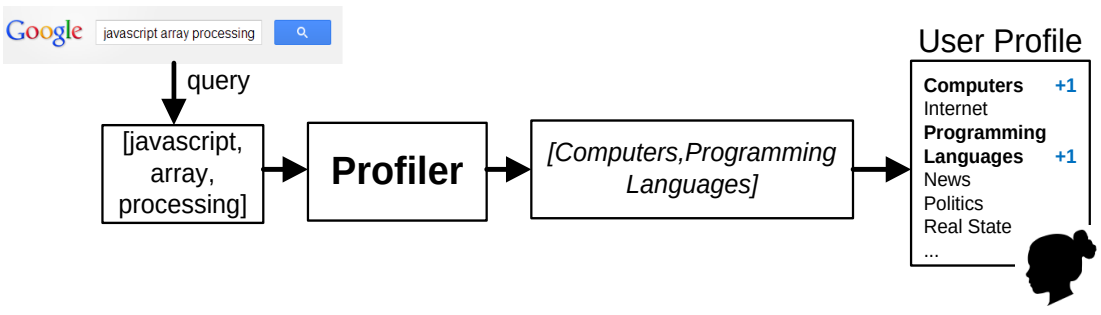

Fig. 6. Profiler architecture.

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[1]. 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. 7.

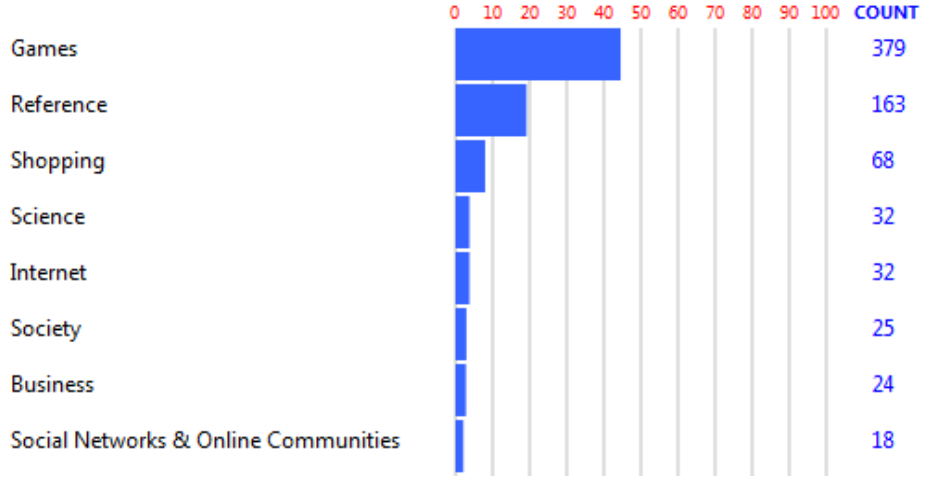

Fig. 7. Histogram representing a basic impression of the user profile.

4.2.4 Identification Module

This module determines the privacy level of a user profile under an identification attack, as explained in section 3.4.1. 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 of a real population. These profiles are obtained from a subset of an AOL [27] 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 [22]), 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

[1] Google Ad Preferences works now with more than 602 categories.

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. 8, the architecture of this module is illustrated.

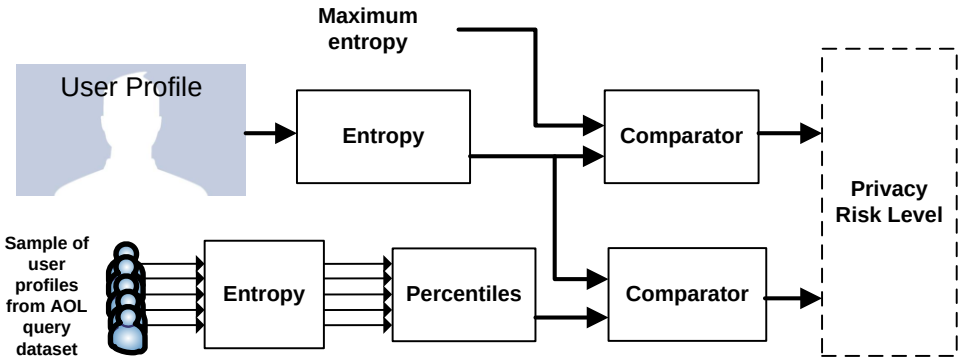

Fig.8. Architecture of identification process.

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. 9). It recreates the attack that would be done by an adversary trying to classify the user within 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.

This classification method is consistent with the metric and the attacker described in the section 3.4.2 and also illustrated in the Fig. 3, 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 [19].

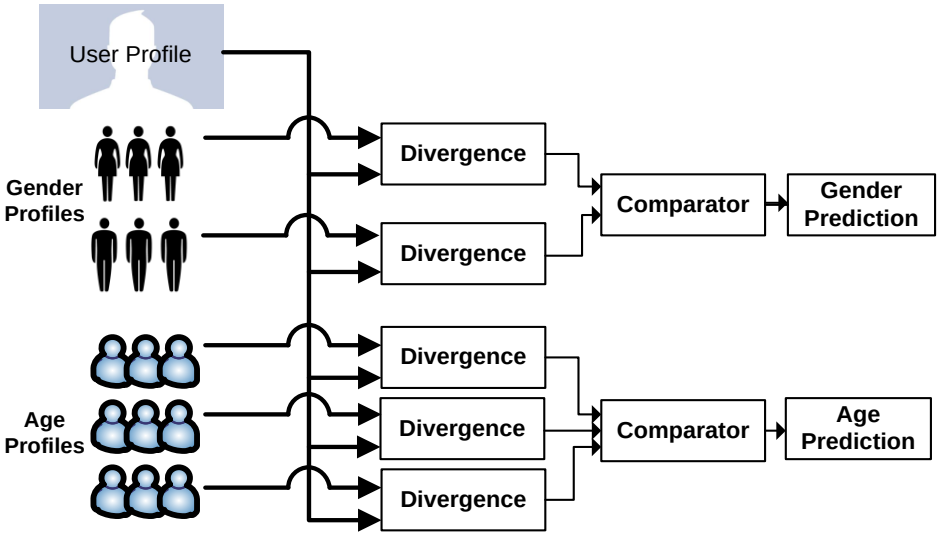

Fig. 9. Architecture of identification module.

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 is created with the objective of measuring the risk level of user's privacy and of showing this level to the user so he can interpret this information. This risk level is calculated according to the considerations analyzed in Chapter 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 [7] extension.

4.3.1 Extension development in Firefox

Firefox is a free and open source web browser developed for Windows, Mac OS and Linux, managed by the Mozilla Corporation and Mozilla Foundation. It uses a Gecko engine to render web pages by implementing current and future web standards.

Even when Firefox is not anymore the most popular web browser [23], it has a very accessible extension system which has contributed to have the largest extension base of all web browsers. Consequently, the documentation available for extension development is abundant, but also is the amount of tools to facilitate coding and debugging.

4.3.1.1 Firefox extensions

Mozilla extensions are small add-ons that add new functionalities to Mozilla applications (Firefox is one of these applications). Extensions allow to add anything to Firefox, from a button for a tool bar to a completely new characteristic, such as the one implemented in this project. Firefox, hence, could be completely personalized to satisfy the user requirements.

Some extensions such as AdBlock Plus, Adnostic and TMN have been mentioned in Chapter 2 and part of their code has been used for this implementation.

When developing Firefox extensions, it is important to have in mind that Firefox structure is closer to a web application than a conventional desktop application [24], because, as seen in Fig. 10, it is pretty similar to web pages using Dynamic HTML.

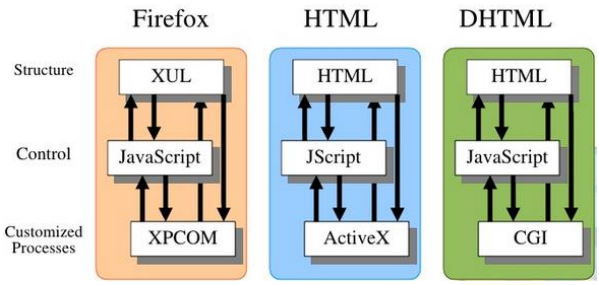

Fig. 10. Firefox architecture compared to a web application's architecture (taken from [24]).

XPCOM, Javascript and XUL are the main modules of Firefox architecture to be accessed in order to give the required functionality to our extension:

- XPCOM, which is a development environment providing features like: component management, file abstraction, object message passing and memory management.
- Javascript, which is used to control all the Firefox components.
- XUL, as a framework of Firefox GUI.

4.3.1.2 Extension developing environment

Firefox extensions are commonly distributed as compressed files using ZIP format and with extension xpi. This xpi file contains some files and directories with the code and content (images, databases) to be used by the extension to accomplish its objective.

For facilitating the Firefox extension development process, a developing environment is built by means of some tools that will help to flexibly manipulate the code and obtain debugging information. The following items describe some of these tools:

- **Development profile.** Mozilla Firefox have the possibility to create profiles. A profile is, basically, a user space where all the browsing information is stored (history, cache files, extensions' code, etc.). Creating a new profile is a very interesting option to isolate the workspace for the developing work.

  To create a new profile, we can use the Profile Manager, accessed from Windows operating system by using the command **`firefox.exe -P`** in the Windows console. There we can indicate the name and location of the new profile.

  To open a new Firefox window using the created profile we can use the command **`firefox.exe -p extension_name`**, from the command line.

- **Development preferences.** These are configuration settings that could be enabled to activate some facilities that will significantly help us in the developing process. In order to access the configuration manager, we have only to write **`about:config`** in the URL window. The following are some of the settings that were activated for our development:

  - **javascript.options.showInConsole = true.** Which logs errors in chrome files to the Error console.
  - **nglayout.debug.disable_xul_cache = true.** Which disables the XUL cache so that changes done to windows and dialogs do not require a browser restart.
  - **devtools.chrome.enabled = true.** Gives access to the Scratchpad tool which allows executing code snippets directly in the chrome context. Changing the context to browser is necessary.
  - **extensions.logging.enabled = true.** This sends more detailed information about installation and update problems to the Error Console.

- **Folder structure and files.** The main tree in which files and directories are organized in the extension is the following (taken from [26]).

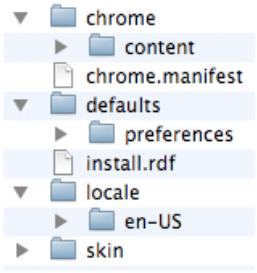

Fig. 11. Folder structure of a Firefox extension.

- **`install.rdf`.** It contains information about the extension: Firefox versions supported, IDs, descriptions and others.
- **`chrome.manifest`.** It is a schema that indicates where the code of every window, toolbar, menu, etc., is located. When a menu or a dialog is created, it should be registered in this file.
- **`chrome` folder.** It essentially contains all the code for the different elements of the extension.
  - **`content/browser.xul`**. It is the file used to override some of the default components of the browser. As an example, a new item for the Tools menu should be coded here.
  - **`content/options.xul`.** If the extension is configurable by means of some preferences, the dialog should be programmed in this file. A corresponding entry has to be registered in the manifest file.
  - **Javascript code.** The js files involving the controlling code of the extension is usually included in the content folder.
- **`defaults` folder.** This folder uses to contain the preferences folder including javascript code to enable (or disable) the preferences or options of the extension.
- **`locale` folder.** This folder may contain the labels used in the extension, with the advantage of being organized by language to facilitate the translation by user configuration.
- **`skin` folder**. This folder contains some images or css files (styles) that will be used for the graphical interface of the extension.

4.3.1.3 Development extensions

Even when some browser development settings have being enabled, some extensions are still capable of helping in the developing work (especially at debugging stages). The following Firefox extensions were installed in the developing environment:

- **Venkman**. It is a Javascript debugger, also available for Thunderbird.
- **Console²**. It provides an enhanced Javascript console.

- **SQLite Manager**. It is used to manage the SQLite databases that Firefox uses to locally store information.
- **DOM Inspector.** It is used to easily inspect or edit any web document or XUL application.
- **Error Console.** It adds a button to quickly access the Firefox Error Console.
- **Restartless Restart.** It adds a button to Firefox to quickly restart the browser window.

4.3.2 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 [7] 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 (as described in Chapter 3) 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. More or less the user profile looks as shown in the Fig. 12.

```
<Category click="0" time="0" count="380" id="Games" xmlns="http://www.mozill
  - <Category click="0" time="0" count="380" id="Video Games">
        <Category click="0" time="0" count="376" id="Online Games"/>
    </Category>
</Category>
<Category click="0" time="0" count="164" id="Reference" xmlns="http://www.mc
    <Category click="0" time="0" count="132" id="Educational Resources"/>
    <Category click="0" time="0" count="15" id="Dictionaries & Encyclopedias"/>
    <Category click="0" time="0" count="1" id="Time & Calendars"/>
    <Category click="0" time="0" count="3" id="Online Directories"/>
</Category>
<Category click="0" time="0" count="68" id="Shopping" xmlns="http://www.mozi
  - <Category click="0" time="0" count="2" id="Flowers Gifts & Greetings">
        <Category click="0" time="0" count="2" id="Gifts"/>
    </Category>
</Category>
<Category click="0" time="0" count="34" id="Internet" xmlns="http://www.mozil
  - <Category click="0" time="0" count="21" id="Web Services">
        <Category click="0" time="0" count="13" id="Search Engines"/>
        <Category click="0" time="0" count="6" id="Affiliate Programs"/>
    </Category>
    <Category click="0" time="0" count="11" id="Online Goodies"/>
    <Category click="0" time="0" count="2" id="Web Portals"/>
</Category>
```

Fig. 12. Excerpt of a user profile as it is stored in the Profile.xml file.

4.3.3 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 in the following groups:

- by age (in years) :
    - 18 to 24,
    - 25 to 34,
    - 35 to 44,
    - 45 to 54,
    - 55 to 64,
    - 65 and more,
- by gender (man or woman)

Section 3.4.2 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. 3, 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* [22]. 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.

4.3.4 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 is the first idea that a user receives about his privacy from our extension. This information bar is made up of an icon and some text fields where the following information is briefly shown:

- Privacy risk level (icon)
- User's profile entropy
- Category where the last user's query was classified

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. 13, a level indicator shows the user how high is his profile's risk level. From white to red, this icons alerts about the privacy risk level, according to the user's profile entropy, which is also displayed.

Fig. 13. Fast privacy information bar.

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

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
  - 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. 14, 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.

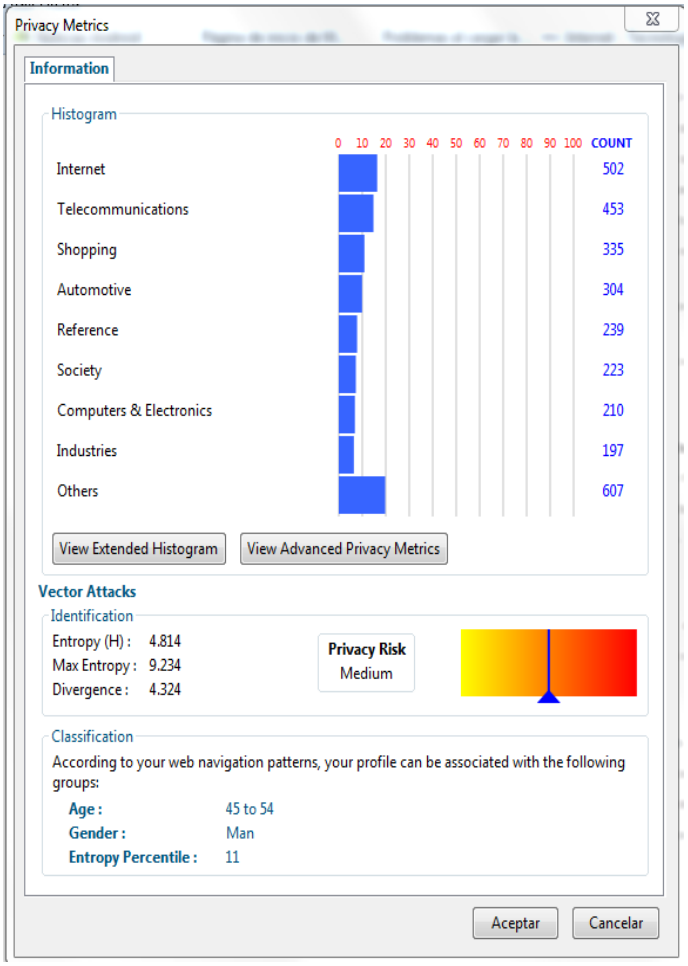

Fig. 14. Main privacy metrics window.

4.3.4.3 Web history import

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. Conclusions and future work

5.1 Conclusions

Considering the big risk that users are facing when browsing the Internet due to aggressive profiling mechanisms used by the main systems providing access to Internet (ISPs, search engines), the creation of an application to determine the user's privacy level in real time, definitely worth it.

The usage of justified metrics could help to solidly evaluate how effective are some privacy protection mechanisms that are already in the landscape.

Privacy is not an isolated concept; the boundaries of what is considered private depend on cultures and individuals. Personal perception and available information will change the way how a user define his privacy. But not having enough information will completely abstract the user from the real risks. That said, data about privacy is essential in order for the user to be aware of how his queries and browsing behavior is revealing very sensitive information about his profile, even when each query or web page does not say much by its own.

Maybe the most serious problem about privacy on the Web is that still many people feel anonymous in Internet, so they do not take any measure to protect their personal profiles. That is why they need information about their privacy, but accessible (visible), and easily interpretable.

This Firefox extension, measures the user's privacy, by using a couple of metrics justified with information theoretic concepts. The user will be able to make reasoned decisions about protecting his privacy when analyzing the calculated risk level or the groups in which he has been classified, or simply by seeing in his histogram a so popular category that could reveal some behavior he does not want to make public.

It is also justifiable that a privacy protection tool should make decisions based on the current user privacy level. So measurement of privacy is not only fundamental for users but also for tools trying to reduce the risk by any mechanism.

The quality of the information on reference profiles and the profiling scheme are essential to effectively simulate an attack that a particular adversary would do. As justified previously, measuring privacy also depends on the adversary model considered for the measuring strategy.

5.2 Future work

Measuring the privacy in the browser opens some interesting doors in the way to enhancing the protection of the user's profile during his navigation on Internet.

Integrating different adversary models, in the sense of implementing diverse categorization schemes (e.g. different hierarchies with more categories), would help the user to have a wider visualization of his privacy even when it will never be complete.

Moreover, it is important to implement or adapt privacy protection tools, such as TrackMeNot, modifying them to work according to the risk level of the user and even to his own personal considerations towards personal information.

For example, when obfuscation of queries is implemented as a privacy protection mechanism, continuous evaluation of apparent and real profiles would allow the user to monitor the privacy gain or reduction of risk he is getting thanks to this mechanism.

Having more complete information about reference population groups would also enrich the analysis of user's privacy, being this also a mechanism of modeling different capabilities of the adversary, when interpreting the privacy measures for the user.

The user profiling in the browser is a component that could also be improved. Capturing more user information, or better interpreting this data would help to have a more detailed user's profile, which is the base of privacy measuring. In this way, the availability of query log datasets, for example, is useful to train categorization engines and to obtain reference population profiles.

6. References

1. Arvind Narayanan y Vitaly Shmatikov, "Robust De-anonymization of Large Sparse Datasets". Security and Privacy, 2008. SP 2008. IEEE Symposium on, C1, 2008.
2. Michael Barbaro and Tom Zeller Jr., "A Face Is Exposed for AOL Searcher No. 4417749". En The New York Times, Technology, URL http://www.nytimes.com/ 2006/08/09/ technology/09aol.html?pagewanted=all, August 2006.
3. Eckersley, Peter. "How unique is your web browser?." Privacy Enhancing Technologies. Springer Berlin Heidelberg, 2010.
4. Eric Pfanner, "Internet Providers in Deal for Tailored Ads". The New York Times, Technology, URL http://www.nytimes.com/2008/02/18/technology/ 18target.html?_r=2&oref=slogin&, February 2008.
5. Katy Hafner, “Google Resists U.S. Subpoena of Search Data”. The New York Times, Technology, URL http://www.nytimes.com/2006/01/20/technology/ 20google.html?_r=1, January 2006.
6. Russia Today, “Google faces the FBI for not disclosing private data of its users”, April 2013.
7. V. Toubiana, D. Boneh, H. Nissenbaum, and S. Barocas, “Adnostic: Privacy Preserving Targeted Advertising ∗”. Proceedings of the 17th Annual Network and Distributed System Security Symposium (NDSS), 2009.
8. Privacy Rights Clearinhouse, “Using the Internet Safely”
9. D. Howe and H. Nissenbaum. “TrackMeNot: resisting surveillance in web search”, 2006. mrl. nyu.edu/˜dhowe/trackmenot/.
10. S. Vi-a, G. News, S. Vi-b, I. Browsing, and I. Explorer, “R E P RIV : Re-Envisioning In-Browser Privacy.”
11. Google Sharing, URL https://addons.mozilla.org/en-us/firefox/addon /googlesharing/
12. Maone, Giorgio. NoScript. Online: http://noscript.net, 2009.
13. Palant, Wladimir: Adblock Plus: Save your time and traffic, http://adblockplus.org/.
14. DoNotTrackMe, URL https://addons.mozilla.org/en-US/firefox/addon/ donottrackplus/?

15. David Rebollo-Monedero, Jordi Forné, and Josep Domingo-Ferrer, “Query Profile Obfuscation by Means of Optimal Query Exchange between Users”. IEEE Trans. Depend., Secure Comput., 2012.
16. J. Parra-Arnau, D. Rebollo-Monedero and J. Forné, “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.
17. J. Parra-Arnau, D. Rebollo-Monedero, J. Forné, J. L. Muñoz y O. Esparza, “Optimal tag suppression for privacy protection in the semantic Web”. Data, Knowl. Eng., vol. 81-82, pp. 46-66, 2012.
18. J. Parra-Arnau, A. Perego, E. Ferrari, J. Forné and D. Rebollo-Monedero, “Privacy-Preserving Enhanced Collaborative Tagging”. IEEE Trans. Knowl. Data Eng., 2012.
19. J. Parra-Arnau, D. Rebollo-Monedero, J. Forné, “Measuring the Privacy of User Profiles in Personalized Information Systems”. Future Generation Computer Systems, 2013.
20. D. Rebollo-Monedero, J. Parra-Arnau, Claudia Diaz and J. Forné, “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.
21. J. Drennan,G. Sullivan and J. Previte, “Privacy, Risk Perception, and Expert Online Behavior: An Exploratory Study of Household End Users”. Journal of Organizational and End User Computing (JOEUC), 18(1), 1-22, 2006.
22. Google Ad Planner, URL https://www.google.com/adplanner/ #audienceBuilder.
23. Browser Stats, W3Schools, http://www.w3schools.com/browsers/ browsers_stats.asp.
24. Mozilla Firefox Extensions Development Tutorial 2009, October FOSS GN09 at Engineering College Bikaner Abhinav Chittora, Google Summer of Code Student, Xiph.org Foundation 2009.
25. Creating an extension, Mozilla Firefox Development Network, https://developer.mozilla.org /es/docs/Creando_una_extensi%C3%B3n
26. How to develop a Firefox extension, http://robertnyman.com/2009/01/24/how-to-develop-a-firefox-extension/

27. G. Pass, A. Chowdhury, C. Torgeson, "A Picture of Search". The First International Conference on Scalable Information Systems, Hong Kong, June, 2006.
28. Brandimarte, Laura, Alessandro Acquisti, and George Loewenstein. "Misplaced Confidences Privacy and the Control Paradox." Social Psychological and Personality Science 4.3 (2013): 340-347.

7. Appendix

7.1 Profiling functions (from Adnostic extension profiler module)

The following are some of the most important functions that are in charge of the profiling process in the Adnostic Firefox extension. They capture words from the browsing process of the user and process those words to get related categories. These categories, and the times of occurrence are employed to build the user profile.

`_createXMLProfile()` – It creates the user profile (XML file), if it does not exist.

`_initProfile()` – It initializes the user profile hierarchy.

`_getMetaKeywords(doc, url)` – Receives the web document as an object and returns the web page metadata as an array of words representing the content of the site.

`_getKeyWords( urlArg, doc, sch )` – Returns a list of categories corresponding to: the content found in a web document or a sent web search.

`_createNewCategory( id )` – Creates a new category to the hierarchical scheme of the user profile.

`_arrayToXMLFile( array )` – Transforms a structured array into an xml file. It basically receives the user profile as an array and writes it structured in the corresponding xml file to be easily read by other functions.

`_updateCategory( category, inc_count, inc_time, inc_click )` – Updates the counters for a category: number of occurrences of the category, visited time and number of clicks. The user profile file is updated according to this increments.

`_profileURL(url, doc, sch)` – Gets a list of categories assigned to an URL and updates the user profile file accordingly.

`_hasBeenTagged(url)` – Verifies if the URL being categorized has been previously processed. Adnostic saves a register of these URLs in the Firefox tagging system.

`_processProfiling: function(evt)` – Executes the profiling function for the current visited web page. This is the function executed by the browser every time a web page loading event is detected, in order to process search queries and HTML content to correspondingly update the user profile.

7.2 Code of the “Privacy Risk” bar

7.2.1 XUL code for the graphic interface

This is XML based code to program the structure of the privacy bar in the browser. This bar shows an icon built of colors to show the privacy risk level of the user. Some other parameters such as the user profile entropy and the last updated category are also displayed.

```
<statusbar>
    <statusbarpanel label="Privacy Risk:" onclick="ADNOSTIC.profiler._drawPrivacy(window)"/>
    <statusbarpanel id="privlevel 1"
              image="chrome://adnostic/content/plomo icon.png"
              class="statusbarpanel-iconic"
          tooltiptext="Privacy Level"/>
    <statusbarpanel id="privlevel 2"
              image="chrome://adnostic/content/plomo icon.png"
              class="statusbarpanel-iconic"
          tooltiptext="Privacy Level"/>
    <statusbarpanel id="privlevel_3"
              image="chrome://adnostic/content/plomo_icon.png"
              class="statusbarpanel-iconic"
          tooltiptext="Privacy Level"/>
    <statusbarpanel id="privlevel_message" label="[Normal]"/>
    <statusbarpanel id="priv_entropy" label="User Profile Entropy: (NA)"
onclick="ADNOSTIC.profiler._drawPrivacy(window)"/>
    <statusbarpanel id="priv_lastcategory" label="Last Category: (NA)" style="font-size: 11px;
color: black" onclick="ADNOSTIC.profiler. drawPrivacy(window)"/>
    <spacer flex="1"/>
</statusbar>
```

7.2.2 Javascript `drawPrivacy` function to fill the Privacy Risk Bar

This function is called from the XUL code of the privacy bar to paint its colors according to the privacy risk level of the user, calculated by using the entropy value of the profile.

```
_drawPrivacy : function(_window) {

        var arr = []
        ADNOSTIC.editor. getPlainProfile(arr)
        var entropy = ADNOSTIC.editor._getTotalEntropy(arr)
        maxEntropy = ADNOSTIC.editor._getMaxEntropy(602)
        if(entropy >= 0 && entropy < (maxEntropy/3)){
            ADNOSTIC.profiler. drawPrivLevel("amarillo","naranja","rojo","High",_window)
        }else if(entropy >= (maxEntropy/3) && entropy < (2*maxEntropy/3)){
            ADNOSTIC.profiler. drawPrivLevel("amarillo","naranja","plomo","Medium",_window)
        } else if (entropy >= (2*maxEntropy/3)){
            ADNOSTIC.profiler._drawPrivLevel("amarillo","plomo","plomo","Low",_window)
        }
        return;

    }
```

7.2.3 Javascript `drawPrivLevel` function to draw the privacy risk level

This function only retrieves the graphical components of the privacy bar to paint them depending on the passed arguments.

```
_drawPrivLevel : function(color1, color2, color3, privtag, _window){

        var priv_message = _window.document.getElementById("privlevel_message");
        priv_message.setAttribute("label",privtag);
        priv_message.setAttribute("onclick","alert('Yaooo')");
        priv_message.parentNode.replaceChild(priv_message,priv_message);
        var container1 = _window.document.getElementById("privlevel_1");
        container1.setAttribute("image","chrome://adnostic/content/"+color1+"_icon.png");
        container1.parentNode.replaceChild(container1,container1)
        var container2 = _window.document.getElementById("privlevel_2");
        container2.setAttribute("image","chrome://adnostic/content/"+color2+"_icon.png");
        container2.parentNode.replaceChild(container2,container2)
        var container3 = _window.document.getElementById("privlevel_3");
        container3.setAttribute("image","chrome://adnostic/content/"+color3+"_icon.png");
        container3.parentNode.replaceChild(container3,container3);
        var container4 = _window.document.getElementById("priv_entropy");
        var arr = [];
        _window.ADNOSTIC.editor._getPlainProfile(arr);
        var entropy = _window.ADNOSTIC.editor._getTotalEntropy(arr).toFixed(3);
        container4.setAttribute("label", "User Profile Entropy: "+ entropy);
             return;
          }
```

7.3 Code for “Privacy Metrics” dialog

This is the XUL and Javascript code to program the dialog showing the privacy risk level of the user.

7.3.1 XUL code for the “Privacy Metrics” graphic interface

The XUL structure of the Privacy Metrics window is showed here. Some of the privacy measuring functions are called here to obtain the numerical results to be interpreted for the user as graphics or comparisons.

```
<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<!DOCTYPE overlay SYSTEM 'chrome://adnostic/locale/adnostic.dtd'>

    <!--persist="screenX screenY width height" -->
<dialog id="privacy-information"
  title="Privacy Metrics"
  onload="ADNOSTIC.editor.loadHistogram();"
  onfocus="ADNOSTIC.editor.loadHistogram();"
  ondialogaccept="TRACKMENOT.optionsTMN.saveOptions()"
  orient="vertical"
  autostretch="always"
  persist="screenX screenY"
  buttons="accept,cancel" flex="1"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">

    <script type="application/x-javascript" src="options.js"/>
    <script type="application/x-javascript" src="utils.js"/>
    <script type="application/x-javascript" src="profiler.js"/>
    <script type="application/x-javascript" src="editor.js"/>
    <script type="application/x-javascript" src="network.js"/>
```

```
<tabbox id="mainTabbox" flex="1" >
    <tabs>
      <tab label="Information" style="font-weight: bold; color: #00547d"/>
    </tabs>

    <tabpanels flex="3">
      <!-- ================ GENERAL TAB ================ -->
      <tabpanel flex="1">
          <vbox>
              <groupbox>
                  <caption id="histogram" label="Histogram" style="background-color: #f8f8f7; color: #00547d"  />
                  <hbox align="center">
                       <!-- ================ IMAGEN HISTOGRAMA ================ -->
                      <svg xmlns="http://www.w3.org/2000/svg" version="1.1" baseProfile="full" width="500" height="300">
                              <text font-size="10px" x="247" y="15" fill="red">0</text>
                            <line x1="250" y1="20" x2="250" y2="290"
                                style="stroke:rgb(0,0,0);stroke-width:0.2"/>
                                <text font-size="10px" x="265" y="15" fill="red">10</text>
                            <line x1="270" y1="20" x2="270" y2="290"
                                style="stroke:rgb(0,0,0);stroke-width:0.2"/>
                                <text font-size="10px" x="285" y="15" fill="red">20</text>
                            <line x1="290" y1="20" x2="290" y2="290"
                                style="stroke:rgb(0,0,0);stroke-width:0.2"/>
                                <text font-size="10px" x="305" y="15" fill="red">30</text>
                            <line x1="310" y1="20" x2="310" y2="290"
                                style="stroke:rgb(0,0,0);stroke-width:0.2"/>
                                <text font-size="10px" x="325" y="15" fill="red">40</text>
                            <line x1="330" y1="20" x2="330" y2="290"
                                style="stroke:rgb(0,0,0);stroke-width:0.2"/>
                                <text font-size="10px" x="345" y="15" fill="red">50</text>
                            <line x1="350" y1="20" x2="350" y2="290"
                                style="stroke:rgb(0,0,0);stroke-width:0.2"/>
                                <text font-size="10px" x="365" y="15" fill="red">60</text>
                            <line x1="370" y1="20" x2="370" y2="290"
                                style="stroke:rgb(0,0,0);stroke-width:0.2"/>
                                <text font-size="10px" x="385" y="15" fill="red">70</text>
                            <line x1="390" y1="20" x2="390" y2="290"
                                style="stroke:rgb(0,0,0);stroke-width:0.2"/>
                                <text font-size="10px" x="405" y="15" fill="red">80</text>
                            <line x1="410" y1="20" x2="410" y2="290"
                                style="stroke:rgb(0,0,0);stroke-width:0.2"/>
                                <text font-size="10px" x="425" y="15" fill="red">90</text>
                            <line x1="430" y1="20" x2="430" y2="290"
                                style="stroke:rgb(0,0,0);stroke-width:0.2"/>
                                <text font-size="10px" x="442" y="15" fill="red">100</text>
                            <line x1="450" y1="20" x2="450" y2="290"
                                style="stroke:rgb(0,0,0);stroke-width:0.2"/>
                                <text style="font-weight: bold" font-size="10px" x="465" y="15" fill="blue">COUNT</text>

                                <text id="h-label-count-cat-1" class="hlabelcount" font-size="12px" x="475" y="35" fill="blue">NA</text>
                                <text id="h-label-count-cat-2" class="hlabelcount" font-size="12px" x="475" y="65" fill="blue">NA</text>
                                <text id="h-label-count-cat-3" class="hlabelcount" font-size="12px" x="475" y="95" fill="blue">NA</text>
                                <text id="h-label-count-cat-4" class="hlabelcount" font-size="12px" x="475" y="125" fill="blue">NA</text>
                                <text id="h-label-count-cat-5" class="hlabelcount" font-size="12px" x="475" y="155" fill="blue">NA</text>
                                <text id="h-label-count-cat-6" class="hlabelcount" font-size="12px" x="475" y="185" fill="blue">NA</text>
                                <text id="h-label-count-cat-7" class="hlabelcount" font-size="12px" x="475" y="215" fill="blue">NA</text>
                                <text id="h-label-count-cat-8" class="hlabelcount" font-size="12px" x="475" y="245" fill="blue">NA</text>
                                <text id="h-label-count-cat-9" class="hlabelcount" font-size="12px" x="475" y="275" fill="blue">NA</text>

                              <g id="h-labels">
                                <text id="h-label-cat-1" class="hlabel" font-size="12px" x="10" y="35" fill="black">(Not Available)</text>
                                <text id="h-label-cat-2" class="hlabel" font-size="12px" x="10" y="65" fill="black">(Not Available)</text>
                                <text id="h-label-cat-3" class="hlabel" font-size="12px" x="10" y="95" fill="black">(Not Available)</text>
                                <text id="h-label-cat-4" class="hlabel" font-size="12px" x="10" y="125" fill="black">(Not Available)</text>
                                <text id="h-label-cat-5" class="hlabel" font-size="12px" x="10" y="155" fill="black">(Not Available)</text>
```

```
                    <text id="h-label-cat-6" class="hlabel" font-size="12px" x="10" y="185" fill="black">(Not Available)</text>
                    <text id="h-label-cat-7" class="hlabel" font-size="12px" x="10" y="215" fill="black">(Not Available)</text>
                    <text id="h-label-cat-8" class="hlabel" font-size="12px" x="10" y="245" fill="black">(Not Available)</text>
                    <text id="h-label-cat-9" class="hlabel" font-size="12px" x="10" y="275" fill="black">Others</text>
                  </g>

                  <g id="h-barras">
                    <rect id="h-barra-cat-1" class="hbarra" x="250" y="20" width="0" height="28"
                    style="fill:rgb(55,100,255);stroke-width:0;stroke:rgb(0,0,0)"/>

                    <rect id="h-barra-cat-2" class="hbarra" x="250" y="50" width="0" height="28"
                    style="fill:rgb(55,100,255);stroke-width:0;stroke:rgb(0,0,0)"/>

                    <rect id="h-barra-cat-3" class="hbarra" x="250" y="80" width="0" height="28"
                    style="fill:rgb(55,100,255);stroke-width:0; stroke:rgb(0,0,0)"/>

                    <rect id="h-barra-cat-4" class="hbarra" x="250" y="110" width="0" height="28"
                    style="fill:rgb(55,100,255);stroke-width:0; stroke:rgb(0,0,0)"/>

                    <rect id="h-barra-cat-5" class="hbarra" x="250" y="140" width="0" height="28"
                    style="fill:rgb(55,100,255);stroke-width:0; stroke:rgb(0,0,0)"/>

                    <rect id="h-barra-cat-6" class="hbarra" x="250" y="170" width="0" height="28"
                    style="fill:rgb(55,100,255);stroke-width:0; stroke:rgb(0,0,0)"/>

                    <rect id="h-barra-cat-7" class="hbarra" x="250" y="200" width="0" height="28"
                    style="fill:rgb(55,100,255);stroke-width:0; stroke:rgb(0,0,0)"/>

                    <rect id="h-barra-cat-8" class="hbarra" x="250" y="230" width="0" height="28"
                    style="fill:rgb(55,100,255);stroke-width:0; stroke:rgb(0,0,0)"/>

                    <rect id="h-barra-cat-9" class="hbarra" x="250" y="260" width="0" height="28"
                    style="fill:rgb(55,100,255);stroke-width:0; stroke:rgb(0,0,0)"/>

                    </g>
            </svg>
            <spring flex="1"/>
        </hbox>
        <spring flex="1"/>
            <hbox width="300px" align="center">
                <button label="View Extended Histogram"
                    oncommand="window.open('chrome://adnostic/content/privacy_extended_histogram.xul','Extended Histogram','chrome,centerscreen');" />
                <button label="View Advanced Privacy Metrics"
                    oncommand="window.open('chrome://adnostic/content/privacy_information_advanced.xul','Advanced Privacy Metrics','chrome,centerscreen');" />

            </hbox>
        <spring flex="1"/>
        </groupbox>

    <hbox align="center">
     <text value="Vector Attacks" style="font-weight: bold; color: #00547d"/>
     <text value="    " class="url"/>
    </hbox>

    <groupbox>
        <caption id="identification" label="Identification" style="background-color: #f8f8f7; color: #00547d"  />

        <hbox width="150px">
            <vbox>
                <label>Entropy (H) :</label>
                <label>Max Entropy :</label>
                <label>Divergence  :</label>
            </vbox>

            <vbox>
```

```
                <label id="pi_entropy_value" value="(NA)"></label>
                <label id="pi_maxentropy_value" value="(NA)"></label>
                <label id="pi_divergence_av_value" value="(NA)"></label>
            </vbox>
            <spacer flex="15"/>
            <vbox>
                <groupbox>
                    <vbox align="center">
                        <label style="font-weight: bold">Privacy Risk</label>
                        <label id="pi_risk_label" pack="center">(NA)</label>
                    </vbox>
                </groupbox>

            </vbox>
            <spacer flex="5"/>
            <vbox width="150px" flex="1">

                <svg xmlns="http://www.w3.org/2000/svg" version="1.1" height="60">
                    <defs>
                        <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
                            <stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1" />
                            <stop offset="100%" style="stop-color:rgb(255,0,0);stop-opacity:1" />
                        </linearGradient>
                    </defs>

                    <rect width="150" height="50" fill="url(#grad1)" />
                    <polygon id="pi_risk_triangle" points="65,55 85,55 75,45" style="fill:none;stroke:black;stroke-width:0"/>
                    <line id="pi_risk_line" x1="75" y1="55" x2="75" y2="0" style="fill:blue;stroke:blue;stroke-width:0"/>

                </svg>

            </vbox>
        </hbox>
    </groupbox>

    <groupbox>
        <caption id="classsification" label="Classification" style="background-color: #f8f8f7; color: #00547d"  />

            <hbox width="500px" align="center">
                <description style="width: 500px">
                        According to your web navigation patterns, your profile can be associated
                        with the following groups:
                </description>
            </hbox>

            <hbox width="500px" align="center">
                <vbox>
                    <label style="margin-left:20px; font-weight: bold; color: #00547d">Age :</label>
                    <label style="margin-left:20px; font-weight: bold; color: #00547d">Gender :</label>
                    <label style="margin-left:20px; font-weight: bold; color: #00547d">Entropy Percentile :</label>
                </vbox>

                <vbox>
                    <label style="margin-left:20px; color: #00547d" id="pi_age" value="(Not Available)"></label>
                    <label style="margin-left:20px; color: #00547d" id="pi_gender" value="(Not Available)"></label>
                    <label style="margin-left:20px; color: #00547d" id="pi_entropy_percentile" value="(Not Available)"></label>
                </vbox>
                <spacer flex="15"/>
            </hbox>


    </groupbox>

</vbox>
```

```
    </tabpanel>
   </tabpanels>

</tabbox>

</dialog>
```

7.3.2 Function to load the user profile histogram

This is the Javscript code of the function called to load the user profile histogram. This function builds the histogram, each time the window is loaded and also each time a click is done over the window.

```
loadHistogram : function()  {
                var win =  _getRunningWindow();
                if (!win ) return;

                var domo = win.ADNOSTIC.profiler._getDocLog()
                var elemento = domo.childNodes[0]
                var nelementos = elemento.childNodes.length;
                sortChildNodsByAttribtue (elemento,'count')

                var etiquetas=document.getElementsByClassName("hlabel");
                var barras=document.getElementsByClassName("hbarra");
                var etiquetas_conteo = document.getElementsByClassName("hlabelcount");
                var arr = [];
                win.ADNOSTIC.editor._getPlainProfile(arr);
                var tcount=win.ADNOSTIC.editor._getSumCount(arr);

                var tcount_1level = 0;
                for(i=0;i<nelementos;i++){
                    tcount_1level = tcount_1level + parseInt(elemento.childNodes[i].getAttribute('count'));
                }

                var tcount_others = 0;
                if(nelementos > 8){
                    for (j=8;j<nelementos;j++){
                        tcount_others = tcount_others + parseInt(elemento.childNodes[j].getAttribute('count'));
                    }
                }
                if(nelementos <= 8) n = nelementos
                else n = 8
                for (i=0;i<n;i++){
                    var cat=elemento.childNodes[i].getAttribute('id');
                    var conteo=elemento.childNodes[i].getAttribute('count');
                    var porcentaje=(parseInt(conteo))*2*100/parseInt(tcount_1level);
```

```
                var textnode=document.createTextNode(cat);
                var textnode_conteo = document.createTextNode(conteo);

                etiquetas[i].replaceChild(textnode,etiquetas[i].childNodes[0]);
                etiquetas_conteo[i].replaceChild(textnode_conteo,etiquetas_conteo[i].childNodes[0]);
                barras[i].setAttribute("width",porcentaje);
            }

            if(nelementos > 8){
                porcentaje_others = tcount_others*2*100/parseInt(tcount_1level);
                barras[8].setAttribute("width",porcentaje_others);
                var textnode_conteo = document.createTextNode(tcount_others);
                etiquetas_conteo[8].replaceChild(textnode_conteo,etiquetas_conteo[8].childNodes[0]);
            }


                var arreglo = [];
                win.ADNOSTIC.editor._getPlainProfile(arreglo);
                var entropy = win.ADNOSTIC.editor._getTotalEntropy(arreglo).toFixed(3);
                var maxentropy = win.ADNOSTIC.editor._getMaxEntropy(602).toFixed(3);

                var domfile = win.ADNOSTIC.profiler._getDomProfile("avprofile.xml");
                var divergence = win.ADNOSTIC.profiler._getDivergence(domfile).toFixed(3);
                var pi_entropy = document.getElementById("pi_entropy_value");
                pi_entropy.setAttribute("value",entropy);
                var pi_maxentropy = document.getElementById("pi_maxentropy_value");
                pi_maxentropy.setAttribute("value",maxentropy);
                var pi_divergence = document.getElementById("pi_divergence_av_value");
                pi_divergence.setAttribute("value",divergence);

                var pi_risklabel = document.getElementById("pi_risk_label");
                if(entropy >= 0 && entropy < (maxentropy/3)){
                    pi_risklabel.setAttribute("value","High");
                }else if(entropy >= (maxentropy/3) && entropy < (2*maxentropy/3)){
                    pi_risklabel.setAttribute("value","Medium");
                } else if (entropy >= (2*maxentropy/3)){
                    pi_risklabel.setAttribute("value","Low");
                }
                risk_level = (win.ADNOSTIC.profiler._getPrivRiskLevel())*150/10;
                var pi_risktriangle = document.getElementById("pi_risk_triangle");
                var pi_riskline = document.getElementById("pi_risk_line");
                pi_risktriangle.setAttribute("points",(risk_level-10)+",55 "+(risk_level+10)+",55 "+(risk_level)+",45");
                pi_risktriangle.setAttribute("style","fill:blue;stroke:black;stroke-width:0")

                pi_riskline.setAttribute("x1",risk_level);
                pi_riskline.setAttribute("x2",risk_level);
                pi_riskline.setAttribute("style","fill:blue;stroke:blue;stroke-width:2")
            var divArray = [];
```

```
var div18to24 = new Object();
div18to24.etiqueta = "18 to 24";
var div25to34 = new Object();
div25to34.etiqueta = "25 to 34";
var div35to44 = new Object();
div35to44.etiqueta = "35 to 44";
var div45to54 = new Object();
div45to54.etiqueta = "45 to 54";
var div55to64 = new Object();
div55to64.etiqueta = "55 to 64";
var div65tomore = new Object();
div65tomore.etiqueta = "65 to +";
```

```
var domfile18to24 = win.ADNOSTIC.profiler. getDomProfile("group1824profile.xml");
var domfile25to34 = win.ADNOSTIC.profiler._getDomProfile("group2534profile.xml");
var domfile35to44 = win.ADNOSTIC.profiler._getDomProfile("group3544profile.xml");
var domfile45to54 = win.ADNOSTIC.profiler._getDomProfile("group4554profile.xml");
var domfile55to64 = win.ADNOSTIC.profiler._getDomProfile("group5564profile.xml");
var domfile65tomore = win.ADNOSTIC.profiler._getDomProfile("group65moreprofile.xml");
var domfilemen = win.ADNOSTIC.profiler. getDomProfile("menprofile.xml");
var domfilewomen = win.ADNOSTIC.profiler._getDomProfile("womenprofile.xml");
```

```
div18to24.valor = win.ADNOSTIC.profiler._getDivergence(domfile18to24).toFixed(3);
div25to34.valor = win.ADNOSTIC.profiler._getDivergence(domfile25to34).toFixed(3);
div35to44.valor = win.ADNOSTIC.profiler._getDivergence(domfile35to44).toFixed(3);
div45to54.valor = win.ADNOSTIC.profiler._getDivergence(domfile45to54).toFixed(3);
div55to64.valor = win.ADNOSTIC.profiler._getDivergence(domfile55to64).toFixed(3);
div65tomore.valor = win.ADNOSTIC.profiler._getDivergence(domfile65tomore).toFixed(3);
```

```
divArray.push(div18to24);
divArray.push(div25to34);
divArray.push(div35to44);
divArray.push(div45to54);
divArray.push(div55to64);
divArray.push(div65tomore);
```

```
divArray.sort(mycomparator);
var pi_age_label = divArray[0].etiqueta;
```

```
var divmen = win.ADNOSTIC.profiler._getDivergence(domfilemen).toFixed(3);
var divwomen = win.ADNOSTIC.profiler._getDivergence(domfilewomen).toFixed(3);
if(divmen < divwomen){
    var pi_gender_label = "Man";
} else {
    var pi_gender_label = "Woman";
}
```

```
                var pi_age = document.getElementById("pi_age");
                pi_age.setAttribute("value",pi_age_label);
                var pi_gender = document.getElementById("pi_gender");
                pi_gender.setAttribute("value",pi_gender_label);

                var percentiles = [];
                percentiles = win.ADNOSTIC.profiler.getListFromFile("percentiles501to1000");
                var percentile = win.ADNOSTIC.profiler.getProfilePercentil(percentiles, entropy);
                var pi_entropy_percentile = document.getElementById("pi_entropy_percentile");
                if(percentile == null){
                    pi_entropy_percentile.setAttribute("value","Out of percentile range");
                } else {
                    pi_entropy_percentile.setAttribute("value",percentile);
                }

                return;
        }
```

7.3.3 Function to load the extended histogram

This function loads the extended histogram of the user profile by showing 20 categories and not only 8 as the previous histogram loaded in the main Privacy Metrics dialog.

```
loadExtendedHistogram : function()   {
                    var win =  _getRunningWindow();
                if (!win ) return;

                var domo = win.ADNOSTIC.profiler._getDocLog()
                var elemento = domo.childNodes[0]
                var nelementos = elemento.childNodes.length;
                sortChildNodsByAttribtue (elemento,'count')

                var etiquetas=document.getElementsByClassName("hlabel");
                var barras=document.getElementsByClassName("hbarra");
                var etiquetas_conteo = document.getElementsByClassName("hlabelcount");
                var arr = [];
                win.ADNOSTIC.editor._getPlainProfile(arr);
                var tcount=win.ADNOSTIC.editor._getSumCount(arr);

                var tcount_1level = 0;
                for(i=0;i<nelementos;i++){
                    tcount_1level = tcount_1level + parseInt(elemento.childNodes[i].getAttribute('count'));
```

```
                }

                for (i=0;i<19;i++){
                    var cat=elemento.childNodes[i].getAttribute('id');
                    var conteo=elemento.childNodes[i].getAttribute('count');
                    var porcentaje=(parseInt(conteo))*2*100/parseInt(tcount_llevel);
                    var textnode=document.createTextNode(cat);
                    var textnode_conteo = document.createTextNode(conteo);
                    etiquetas[i].replaceChild(textnode,etiquetas[i].childNodes[0]);
                    etiquetas_conteo[i].replaceChild(textnode_conteo,etiquetas_conteo[i].childNodes[0]);
                    barras[i].setAttribute("width",porcentaje);
                }


        }
```

7.3.4 Function to load the Privacy Advanced Metrics dialog

This function loads more advanced metrics to a secondary browser dialog.

```
loadPrivacyAdvancedMetrics : function() {
            var win =  _getRunningWindow();
                if (!win ) return;
                    var arreglo = [];
                    win.ADNOSTIC.editor._getPlainProfile(arreglo);
                    var entropy = win.ADNOSTIC.editor._getTotalEntropy(arreglo).toFixed(3);
                    var maxentropy = win.ADNOSTIC.editor._getMaxEntropy(602).toFixed(3);

                    var domfile = win.ADNOSTIC.profiler. getDomProfile("avprofile.xml");
                    var divergence = win.ADNOSTIC.profiler._getDivergence(domfile).toFixed(3);

                    var pia_entropy = document.getElementById("pia_entropy_value");
                    pia_entropy.setAttribute("value",entropy);

                    var pia divergence = document.getElementById("pia_divergence_value");
                    pia_divergence.setAttribute("value",divergence);

                    entropy_level = entropy*100/maxentropy;
                    var pia_entropy_triangle = document.getElementById("pia_entropy_triangle");
                    pia_entropy_triangle.setAttribute("points",(entropy_level+20-5)+",25 "+(entropy_level+20+5)+",25
"+(entropy level+20)+",15");
                    pia_entropy_triangle.setAttribute("style","fill:red;stroke:black;stroke-width:0")

                var percentiles = [];
                percentiles = win.ADNOSTIC.profiler.getListFromFile("percentiles501to1000");
                var percentile = win.ADNOSTIC.profiler.getProfilePercentil(percentiles, entropy);
                var pia_entropy_percentile = document.getElementById("pia_percentile_value");
```

```
                pia_entropy_percentile.setAttribute("value",percentile);

                var pia_risk_triangle = document.getElementById("pia_risk_triangle");
                pia_risk_triangle.setAttribute("points",(100-percentile+20-5)+",25 "+(100-percentile+20+5)+",25 "+(100-
percentile+20)+",15");
                pia_risk_triangle.setAttribute("style","fill:red;stroke:black;stroke-width:0")

                var divArray = [];
                var div18to24 = new Object();
                div18to24.etiqueta = "18 to 24";
                var div25to34 = new Object();
                div25to34.etiqueta = "25 to 34";
                var div35to44 = new Object();
                div35to44.etiqueta = "35 to 44";
                var div45to54 = new Object();
                div45to54.etiqueta = "45 to 54";
                var div55to64 = new Object();
                div55to64.etiqueta = "55 to 64";
                var div65tomore = new Object();
                div65tomore.etiqueta = "65 to +";

                var domfile18to24 = win.ADNOSTIC.profiler._getDomProfile("group1824profile.xml");
                var domfile25to34 = win.ADNOSTIC.profiler._getDomProfile("group2534profile.xml");
                var domfile35to44 = win.ADNOSTIC.profiler._getDomProfile("group3544profile.xml");
                var domfile45to54 = win.ADNOSTIC.profiler._getDomProfile("group4554profile.xml");
                var domfile55to64 = win.ADNOSTIC.profiler. getDomProfile("group5564profile.xml");
                var domfile65tomore = win.ADNOSTIC.profiler._getDomProfile("group65moreprofile.xml");
                var domfilemen = win.ADNOSTIC.profiler._getDomProfile("menprofile.xml");
                var domfilewomen = win.ADNOSTIC.profiler._getDomProfile("womenprofile.xml");

                div18to24.valor = win.ADNOSTIC.profiler._getDivergence(domfile18to24).toFixed(3);
                div25to34.valor = win.ADNOSTIC.profiler._getDivergence(domfile25to34).toFixed(3);
                div35to44.valor = win.ADNOSTIC.profiler._getDivergence(domfile35to44).toFixed(3);
                div45to54.valor = win.ADNOSTIC.profiler._getDivergence(domfile45to54).toFixed(3);
                div55to64.valor = win.ADNOSTIC.profiler._getDivergence(domfile55to64).toFixed(3);
                div65tomore.valor = win.ADNOSTIC.profiler._getDivergence(domfile65tomore).toFixed(3);

                divArray.push(div18to24);
                divArray.push(div25to34);
                divArray.push(div35to44);
                divArray.push(div45to54);
                divArray.push(div55to64);
                divArray.push(div65tomore);

                divArray.sort(mycomparator);

                var anterior;
                divArray[0].posiciony=32;
```

```
                for(i=1;i<divArray.length;i++){
                    if((divArray[i].valor-divArray[i-1].valor) > (divArray[5].valor - divArray[0].valor)*35/380) {
                        divArray[i].posiciony=32;
                        anterior=32;
                    } else {
                        if(anterior == 1) {
                            divArray[i].posiciony=32;
                            anterior = 32;
                        } else {
                            divArray[i].posiciony=42;
                            anterior=1;
                        }
                    }
                }

                var pia_div_age_norm = divArray[5].valor - divArray[0].valor;

                var pia_div_18to24_relative = (div18to24.valor - divArray[0].valor)*380/pia_div_age_norm;
                var pia_div_25to34_relative = (div25to34.valor - divArray[0].valor)*380/pia_div_age_norm;
                var pia_div_35to44_relative = (div35to44.valor - divArray[0].valor)*380/pia_div_age_norm;
                var pia_div_45to54_relative = (div45to54.valor - divArray[0].valor)*380/pia_div_age_norm;
                var pia_div_55to64_relative = (div55to64.valor - divArray[0].valor)*380/pia_div_age_norm;
                var pia_div_65tomore_relative = (div65tomore.valor-divArray[0].valor)*380/pia_div_age_norm;

                var pia_div_18to24_triangle = document.getElementById("pia_div_18to24_triangle");
                    pia_div_18to24_triangle.setAttribute("points",(40+pia_div_18to24_relative)+",25
            "+(50+pia_div_18to24_relative)+",25 "+(45+pia_div_18to24_relative)+",15");
                    pia_div_18to24_triangle.setAttribute("style","fill:red;stroke:black;stroke-width:0")
                var pia_div_label_18to24 = document.getElementById("pia_div_label_18to24");
                var textnode1=document.createTextNode(div18to24.etiqueta);
                pia_div_label_18to24.replaceChild(textnode1,pia_div_label_18to24.childNodes[0]);
                pia_div_label_18to24.setAttribute("x",45+pia_div_18to24_relative-15);
                pia_div_label_18to24.setAttribute("y",divArray[divArray.indexOf(div18to24)].posiciony);

                var pia_div_25to34_triangle = document.getElementById("pia_div_25to34_triangle");
                                pia_div_25to34_triangle.setAttribute("points",(40+pia_div_25to34_relative)+",25
                   "+(50+pia div 25to34 relative)+",25 "+(45+pia div 25to34 relative)+",15");
                    pia_div_25to34_triangle.setAttribute("style","fill:red;stroke:black;stroke-width:0")
                var pia_div_label_25to34 = document.getElementById("pia_div_label_25to34");
                var textnode2=document.createTextNode(div25to34.etiqueta);
                pia_div_label_25to34.replaceChild(textnode2,pia_div_label_25to34.childNodes[0]);
                pia_div_label_25to34.setAttribute("x",45+pia_div_25to34_relative-15);
                pia_div_label_25to34.setAttribute("y",divArray[divArray.indexOf(div25to34)].posiciony);

                var pia_div_35to44_triangle = document.getElementById("pia_div_35to44_triangle");
                pia_div_35to44_triangle.setAttribute("points",(40+pia_div_35to44_relative)+",25 "+(50+pia_div_35to44_relative)+",25
"+(45+pia_div_35to44_relative)+",15");
                pia_div_35to44_triangle.setAttribute("style","fill:red;stroke:black;stroke-width:0")
```

```
                var pia_div_label_35to44 = document.getElementById("pia_div_label_35to44");
                var textnode3=document.createTextNode(div35to44.etiqueta);
                pia_div_label_35to44.replaceChild(textnode3,pia_div_label_35to44.childNodes[0]);
                pia_div_label_35to44.setAttribute("x",45+pia_div_35to44_relative-15);
                pia_div_label_35to44.setAttribute("y",divArray[divArray.indexOf(div35to44)].posiciony);

                var pia_div_45to54_triangle = document.getElementById("pia_div_45to54_triangle");
                pia_div_45to54_triangle.setAttribute("points",(40+pia_div_45to54_relative)+",25 "+(50+pia_div_45to54_relative)+",25
"+(45+pia_div_45to54_relative)+",15");
                pia_div_45to54_triangle.setAttribute("style","fill:red;stroke:black;stroke-width:0")
                var pia_div_label_45to54 = document.getElementById("pia_div_label_45to54");
                var textnode4=document.createTextNode(div45to54.etiqueta);
                pia_div_label_45to54.replaceChild(textnode4,pia_div_label_45to54.childNodes[0]);
                pia_div_label_45to54.setAttribute("x",45+pia_div_45to54_relative-15);
                pia_div_label_45to54.setAttribute("y",divArray[divArray.indexOf(div45to54)].posiciony);

                var pia_div_55to64_triangle = document.getElementById("pia_div_55to64_triangle");
                pia_div_55to64_triangle.setAttribute("points",(40+pia_div_55to64_relative)+",25 "+(50+pia_div_55to64_relative)+",25
"+(45+pia_div_55to64_relative)+",15");
                pia_div_55to64_triangle.setAttribute("style","fill:red;stroke:black;stroke-width:0")
                var pia_div_label_55to64 = document.getElementById("pia_div_label_55to64");
                var textnode5=document.createTextNode(div55to64.etiqueta);
                pia_div_label_55to64.replaceChild(textnode5,pia_div_label_55to64.childNodes[0]);
                pia_div_label_55to64.setAttribute("x",45+pia_div_55to64_relative-15);
                pia_div_label_55to64.setAttribute("y",divArray[divArray.indexOf(div55to64)].posiciony);

                var pia_div_65tomore_triangle = document.getElementById("pia_div_65tomore_triangle");
                pia_div_65tomore_triangle.setAttribute("points",(40+pia_div_65tomore_relative)+",25
"+(50+pia_div_65tomore_relative)+",25 "+(45+pia_div_65tomore_relative)+",15");
                pia_div_65tomore_triangle.setAttribute("style","fill:red;stroke:black;stroke-width:0")
                var pia_div_label_65tomore = document.getElementById("pia_div_label_65tomore");
                var textnode6=document.createTextNode(div65tomore.etiqueta);
                pia_div_label_65tomore.replaceChild(textnode6,pia_div_label_65tomore.childNodes[0]);
                pia_div_label_65tomore.setAttribute("x",45+pia_div_65tomore_relative-15);
                pia_div_label_65tomore.setAttribute("y",divArray[divArray.indexOf(div65tomore)].posiciony);

                var divmen = win.ADNOSTIC.profiler. getDivergence(domfilemen).toFixed(3);
                var divwomen = win.ADNOSTIC.profiler._getDivergence(domfilewomen).toFixed(3);

                var pia_div_18to24 = document.getElementById("pia_div_18to24");
                pia_div_18to24.setAttribute("value",div18to24.valor);
                var pia_div_25to34 = document.getElementById("pia_div_25to34");
                pia_div_25to34.setAttribute("value",div25to34.valor);
                var pia_div_35to44 = document.getElementById("pia_div_35to44");
                pia_div_35to44.setAttribute("value",div35to44.valor);
                var pia_div_45to54 = document.getElementById("pia_div_45to54");
                pia_div_45to54.setAttribute("value",div45to54.valor);
                var pia_div_55to64 = document.getElementById("pia_div_55to64");
```

```
                pia_div_55to64.setAttribute("value",div55to64.valor);
                var pia_div_65tomore = document.getElementById("pia_div_65tomore");
                pia_div_65tomore.setAttribute("value",div65tomore.valor);

                var pia_div_man = document.getElementById("pia_div_man");
                pia_div_man.setAttribute("value",divmen);
                var pia_div_woman = document.getElementById("pia_div_woman");
                pia_div_woman.setAttribute("value",divwomen);
                return;
        }
```

7.4 Code for Privacy Measuring

7.4.1 Functions to get the entropy of the user profile

These functions are the ones that calculate the values of metrics used to determine the user privacy level.

```
_getTotalEntropy : function (arr){
            sum = 0
            for(i in arr){
                  sum = sum + parseInt(arr[i].getAttribute("count"))
            }
            entr total = 0
            for(i in arr) {
              entr_total = entr_total +
          ADNOSTIC.editor._calcEntropy(parseInt(arr[i].getAttribute("count")) , sum)
              }
              return entr_total;
        }


_calcEntropy : function(count, sum){
     var entr;
     if(count == 0){
           entr = 0;
           return entr;
     } else {
           entr = (count/sum)*Math.log(sum/count)/Math.log(2)
           return entr;
     }
}
```

7.4.2 Function to get the divergence between the user profile and a reference profile

```
_getDivergence : function (popprofile_dom){
    userprofile_dom = ADNOSTIC.profiler._getDocLog()

    plain_user_profile = []
    ADNOSTIC.editor._recurseDomChildren(userprofile_dom.childNodes[0],plain_user_profile)
    totalcount_user = ADNOSTIC.editor._getSumCount(plain_user_profile)

    plain_pop_profile = []
    ADNOSTIC.editor._recurseDomChildren(popprofile_dom.childNodes[0],plain_pop_profile)
    totalcount_pop = ADNOSTIC.editor._getSumCount(plain_pop_profile)

    divergencia = 0
    for(i=0; i<plain user profile.length; i++){
        c_u =  plain_user_profile[i].getAttribute("count")
        id_u = plain_user_profile[i].getAttribute("id")
        p_u = parseInt(c_u)/totalcount_user
        if(popprofile_dom.getElementById(id_u) == null)
              alert("Error: Category " + id_u + " does not exist in your group profiles.
            Check the xml files")
        c_p = popprofile_dom.getElementById(id_u).getAttribute("count")
        p_p = parseInt(c_p)/totalcount_pop
```

```
        if(c_u == 0){
            divergencia = divergencia + 0;
        } else {
            divergencia = divergencia + p_u*(Math.log(p_u/p_p))/Math.log(2)
        }
    }
    return divergencia;
  }
```

7.4.3 Function to get the privacy risk level of the user profile

Having the entropy of a user profile, this function obtains the corresponding privacy risk level.

```
_getPrivRiskLevel : function() {
        var arr = []
        ADNOSTIC.editor._getPlainProfile(arr)
        var entropy = ADNOSTIC.editor._getTotalEntropy(arr)
        maxEntropy = ADNOSTIC.editor. getMaxEntropy(602)
        if(entropy >= (maxEntropy*9/10) && entropy <= maxEntropy){
            return 1;
        }else if(entropy >= (maxEntropy*8/10) && entropy < (maxEntropy*9/10)){
            return 2;
        } else if(entropy >= (maxEntropy*7/10) && entropy < (maxEntropy*8/10)){
            return 3;
        } else if(entropy >= (maxEntropy*6/10) && entropy < (maxEntropy*7/10)){
            return 4;
        } else if(entropy >= (maxEntropy*5/10) && entropy < (maxEntropy*6/10)){
            return 5;
        } else if(entropy >= (maxEntropy*4/10) && entropy < (maxEntropy*5/10)){
            return 6;
        } else if(entropy >= (maxEntropy*3/10) && entropy < (maxEntropy*4/10)){
            return 7;
        } else if(entropy >= (maxEntropy*2/10) && entropy < (maxEntropy*3/10)){
            return 8;
        } else if(entropy >= (maxEntropy*1/10) && entropy < (maxEntropy*2/10)){
            return 9;
        } else if(entropy >= (maxEntropy*0/10) && entropy < (maxEntropy*1/10)){
            return 10;
        }
    }
```

7.4.4 Function to get the percentile corresponding to a value of entropy

This function retrieves the percentile where a value of entropy is located. The percentile ranges are previously calculated.

```
function _getProfilePercentil (arreglo, valor) {
        if(valor <= arreglo[0]) return 1;
        if(valor >= arreglo[arreglo.length-1]) return 100;
        var lo = 0, hi = arreglo.length-1;
```

```
        while (hi - lo > 1) {
            var mid = Math.round((lo + hi)/2);
            if (arreglo[mid] <= valor) {
                lo = mid;
            } else {
                hi = mid;
            }
        }
        if (arreglo[lo] == valor) hi = lo;
        return arreglo.indexOf(arreglo[hi]);
    }
```

7.5 Code for profile object manipulation

7.5.1 Function to get a list of strings in a file as an array of strings

This function gets the lines of a file as elements of an array to further processing. This is used for retrieving a list of queries from a text file.

```
function _getListFromFile(archivo){

            var id = "adnostic@extension";
        var file = Components.classes["@mozilla.org/file/directory_service;1"]
     .getService(Components.interfaces.nsIProperties).get("ProfD",
     Components.interfaces.nsIFile);
            file.append('extensions');
            file.append(id);
     file.append(archivo);
     var istream = Components.classes["@mozilla.org/network/file-input-stream;1"]
          .createInstance(Components.interfaces.nsIFileInputStream);
        istream.init(file, 0x01, 0444, 0);
        istream.QueryInterface(Components.interfaces.nsILineInputStream);
        arreglo = []
        var line = {}, hasmore;
        do {
            hasmore = istream.readLine(line);
            arreglo.push(line.value);
        } while(hasmore);
            istream.close();
            return arreglo;
}
```

7.5.2 Function to get the most popular category from the user profile

This function is an example of how the user profile is retrieved to order its nodes (categories) from the most popular category to the least popular one.

```
_getFirstCategory : function() {
          var domo = ADNOSTIC.profiler._getDocLog()
          var elemento = domo.childNodes[0]
          sortChildNodsByAttribtue (elemento,'count')
          var hijos = elemento.childNodes
          if(hijos.length>0){
                    primero = elemento.childNodes[0];
```

```
                    return primero;
            } else {return null}
```

7.5.3 Function to get the user profile as a DOM object

This is another function used to get the file user profile as a Javascript object (a DOM object) in such a way that its information can be easily read and manipulated.

```
_getDomProfile : function (pfile){
    var file_profile = ADNOSTIC.utils._getProfileDir().clone();
    file_profile.append("extensions");
    file_profile.append("adnostic@extension")
    file_profile.append(pfile)
    profile dom = ADNOSTIC.utils._getDomRequestFromFile(file_profile);
    return profile_dom;
}
```

7.5.4 Function to get the hierarchical user profile as a plain list of objects

This function gets the user profile from the corresponding XML file to convert it into a plain array.

```
_getPlainProfile : function (arr){
            var domo = ADNOSTIC.profiler. getDocLog()
            var elemento = domo.childNodes[0]
            ADNOSTIC.editor._recurseDomChildren(elemento, arr);
}
```

7.6 Code for Firefox History import process and profiling

This function gets the titles of all the web pages registered in the Firefox history to profile them. The process of importing and profiling this information allows the user to have an initial profile, which can be used to have a measure of privacy immediately after the user has installed the extension.

```
setProfileFromFirefoxHistory : function() {
            var win =  _getRunningWindow();
            if (!win ) return;
            win.ADNOSTIC.options.resetProfile();

            Cc = Components.classes;
            Ci = Components.interfaces;
            var history = Cc["@mozilla.org/browser/nav-history-service;1"]
              .getService(Ci.nsINavHistoryService);

            var query = history.getNewQuery();
            query.searchTerms = "http";

            var result = history.executeQuery(query, history.getNewQueryOptions());

            var resultContainerNode = result.root;

            resultContainerNode.containerOpen = true;
            for (var i=0; i < resultContainerNode.childCount; ++i) {
              var childNode = resultContainerNode.getChild(i);
              var title = childNode.title;
              if(title != null){
                  win.ADNOSTIC.profiler._profileDelicTitle(title);
              }
            }

            dbHandler = new win.PrivDbHandler()
            var entries = dbHandler.getAllEntries();
            for (var j=0; j<entries.length ; j++){
                if(entries[j].value != null){
                    win.ADNOSTIC.profiler._profileDelicTitle(entries[j].value)
                }
            }
            alert("Your profile has been updated");
```