ProfileMeNot: Achieving Privacy in Online Behavioural Advertising through Obfuscation

by

Seyyed Mohammad Hadi Sajjadpour

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master of Computer Science

In

School Of Computer Science

Carleton University
Ottawa, Ontario

Abstract

As users browse the Web, there are companies that track them across different websites. By tracking users, they infer their interests and behavior. The type of information they infer can be as detailed as menopause, getting pregnant, repairing bad credit and debt relief [2]. They predominantly use this information to target users with ads tailored to their interests. Studies show that the majority of Americans are concerned about their privacy with respect to these types of ads and also being tracked across websites. There have been several solutions proposed to address the privacy concerns of users. In this work, we investigate these proposed solutions and discuss their effectiveness. We conclude that none of the proposed solutions comprehensively address privacy concerns or are not widely adapted due to the fact they depend on other parties to change. We do not aim to kill online advertising, as it is a key factor to much of the free content on the Web. Many of the existing solutions block online advertising, while at the same time not being able to completely block tracking. We propose a novel solution, achieving privacy across the Web by using obfuscation, and hope that our solution will encourage the online advertising industry to adapt a guaranteed privacy preserving model. We developed our idea as a Firefox add-on and called it ProfileMeNot. ProfileMeNot is a comprehensive solution that achieves privacy by visiting websites on behalf of the user to skew the interests they infer from them. ProfileMeNot works solely on the client's side and does not require any other party to operate. ProfileMeNot was built on top of an already existing add-on called TrackMeNot. We show that ProfileMeNot can simulate a user and fool the trackers.

Acknowledgements

As the completion of this thesis signals the completion of my Masters, I would like to thank everyone who helped me throughout my program. I would like to thank the Ottawa community for being a warm home for the past few years. In particular, I would like to thank my parents and siblings who supported me all along my thesis. I would also like to thank and express my appreciation at help and guidance given to me by my supervisors, Professor Evangelos Kranakis and Professor Carlisle Adams. I would like to thank Abdallah Douha for designing ProfileMeNot's logo and all the volunteers who helped at the different phases of the ProfileMeNot experiments. Additionally, I would like to thank Mohamed Suleman from the Carleton University School of Journalism for editing my thesis. Lastly, I would like to acknowledge my lab partner Eduardo Pacheco for helping me out with the formatting of my thesis.

Table of Contents

Abstract ..... **ii**

Acknowledgements ..... **iii**

Table of Contents ..... **iv**

List of Tables ..... **ix**

List of Figures ..... **x**

List of Appendices ..... **xii**

List of Acronyms ..... **xiii**

1 Introduction **14**

1.1 Motivation, Problem Statement, and Summary of Contributions **15**

1.2 Organization **16**

2 Online Behavioral Advertising, Tracking and Privacy, a Background **18**

2.1 Targeted Advertising, Third-party Tracking and Recommender Systems **18**

2.1.1 Contextual Advertising 19

2.1.2 Online Behavioral Advertising, Analytics Services and Third-Party Tracking 19

2.1.3 Social Media Targeting 21

2.1.4 Mobile Advertising and Location-based Services 22

2.1.5 Location-Based Advertising 24

2.1.6 Recommender Systems 24

2.2 Privacy 25

2.2.1 Privacy, Profiling and Personal Information 25

2.2.2 Philosophy and Arguments for Data Collection 27

2.2.3 Type of Information Gathered, Sold and Attack Model 29

2.2.4 Statistics on User Concerns 30

2.2.5 Privacy and the Law 31

2.3 Conclusion 34

3 Online Advertising and Tracking Technologies 36

3.1 Online Advertising 36

3.1.1 Online Behavioral Advertising, A History 37

3.1.2 Cookies as a Privacy Risk 38

3.1.3 Development of the Online Advertising Business Model 42

3.1.4 Third-Party Content in First Parties 43

3.2 Emerging Tracking Technologies 45

3.2.1 Flash Cookies 46

3.2.2 HTML5 Storage 51

3.2.3 HTTP ETag 52

3.2.4 Fingerprinting Browsers 53

3.2.5 Convergence of Anonymity, Pseudonymity and Identifiability 55

3.3 Conclusion 56

4 Existing Solutions to OBA and Third-Party Tracking 58

4.1 Opt-outs and Self-Regulation 58

4.1.1 Opt-outs 58

4.1.2 Self-Regulation 59

4.1.2.1 Online Privacy Alliance 60
4.1.2.2 Network Advertising Initiative and Digital Advertising Alliance 60
4.1.3 Do Not Track 62
4.2 P3P: The Platform for Privacy Preferences 64
4.3 Blocking 66
4.3.1 Complete Blocking 67
4.3.2 Blacklisted Blocking and Partial Cookie Blocking 68
4.3.3 Private Browsing Mode 69
4.4 Proxy-based Anonymizers 70
4.5 Solutions that allow both Privacy and OBA 71
4.5.1 Adnostic 71
4.6 Conclusion 74
5 Obfuscation, TrackMeNot and ProfileMeNot 75
5.1 What is Obfuscation? 76
5.2 TrackMeNot 77
5.2.1 Dynamic Query List 78
5.2.2 Selective Click-through 80
5.2.3 Real-Time Search Awareness 80
5.2.4 Live Header Maps 81
5.2.5 Burst-Mode Queries 81
5.2.6 TMN Control Panel 82
5.2.7 Implementation Details 82
5.3 ProfileMeNot 84
5.3.1 ProfileMeNot, Overview 85

5.3.2 Dynamic Query List 85
5.3.3 Burst-Mode 85
5.3.4 ProfileMeNot General Filters 86
5.3.5 Page Visit Algorithms 87
5.3.5.1 Algorithm 1, ProfileMeNot Basic 88
5.3.5.2 Algorithm 2, ProfileMeNot Ultimate 89
5.3.5.2.1 Algorithm 2.1 90
5.3.5.2.2 User Page Revisitation Habits 90
5.3.5.2.3 Algorithm 2.2 93
5.3.5.2.4 Algorithm 2, Selecting Links to Add to Lists, and User Browsing Habits 96
5.3.6 Strengths and Weaknesses of ProfileMeNot 101
5.3.6.1 Strengths 101
5.3.6.2 Potential Weaknesses 103
5.4 Conclusion 105
6 Experiments, Results and Analysis 106
6.1 Experiment Preliminaries 106
6.1.1 Frequency Rates 106
6.1.2 Volunteer Selection and breakdown 107
6.1.3 Google Ads Preferences and Taxonomy 108
6.1.4 Experiment Steps 109
6.2 Comparison Methodology and Results 114
6.2.1 Comparison Methodology 114
6.3 Results 116
6.3.1 Algorithm 1 Results 119
6.3.2 Algorithm 2 Results 120

6.3.3 Commonalities Between Both Algorithms 121
6.3.4 Differences Between Both Algorithms 122
6.3.5 Expectations and effects on OBA 122
6.3.6 Accuracy of Results 123
6.4 Conclusion 125
7 Conclusion and Future Work 126
7.1 Future work 127
List of References 129
Appendices 136
A. Google Ads Preferences Root Categories 136
B. ProfileMeNot Instructions 137
B.1 First batch of instructions 137
B.2 Second batch of instructions 140

List of Tables

Table 1: Comparison of findings in different studies with respect to Flash cookies 50

Table 2: Comparison of tracking technologies.....52

Table 3: Browser fingerprint elements 54

Table 4: List of variables and notation used in Algorithms 94

Table 5: Users Web browsing behavior habits by percentage.....97

Table 6: Size of lists 98

List of Figures

Figure 1: Sample Google ads preferences 41

Figure 2: Example of an iFrame and how it learns about its referrer 45

Figure 3: Google Chrome`s Incognito mode message upon opening a new Incognito window ... 70

Figure 4: Adnostic architecture 73

Figure 5: Query seed list [12] 79

Figure 6: Query seed list from Figure 5 having evolved after a few weeks [12] 79

Figure 7: Amount of bandwidth used by ProfileMeNot in megabytes 104

Figure 8: Average number of days ProfileMeNot was used by volunteers in Algorithm 1 with a frequency of 120 queries per hour 112

Figure 9: Average number of days ProfileMeNot was used by volunteers in Algorithm 1 with a frequency of 60 queries per hour 112

Figure 10: Average number of days ProfileMeNot was used by volunteers in Algorithm 2 with a frequency of 120 queries per hour 113

Figure 11: Average number of days ProfileMeNot was used by volunteers Algorithm 2 with a frequency of 60 queries per hour 113

Figure 12: Average percent of noise in Algorithm 1 with an average frequency rate of 120 queries per hour 116

Figure 13: Average percent of noise in Algorithm 1 with an average frequency rate of 60 queries per hour 117

Figure 14: Average percent of noise in c Algorithm 2 with average frequency rate of 120 queries per hour 117

Figure 15: Average percent of noise in Algorithm 2 with average frequency rate of 60 queries per minute 118
Figure 16: Comparison of algorithms with different frequencies in the first batch 118
Figure 17: Comparison of algorithms with different frequencies in the second batch. The third and sixth columns show the difference in percentage change of each. 119

List of Appendices

A. Google Ads Preferences Root Categories…...……135

B. ProfileMeNot. ……...…….……135

B.1 First Step of the Experiment Instructions……135

B.2 Second Step of the Experiment Instructions……138

List of Acronyms

OBA..... Online Behavioral Advertising

TMN..... TrackMeNot

SMT..... Social Media Targeting

ROI..... Return on Investment

PIPEDA.... Personal Information Protection and Electronic Documents Act

NAI..... Network Advertising Initiative

DNT..... Do Not Track

FTC..... Federal Trade Commission

GNN..... Global Network Navigator

IETF..... Internet Engineering Task Force

NWG..... Network Working Group

RFC..... Request For Comments

CPM..... Cost per Mille

LSO..... Local Shared Objects

ETag..... Entity Tag

OPA..... Online Privacy Alliance

PII..... Personally Identifiable Information

DAA..... Digital Advertising Alliance

DOM..... Document Object Model

TOR..... The Onion Router

RAF..... Royal Air Force

1 Introduction

The goal of advertising is to persuade an audience to take or continue to take a particular action. As media technology has evolved, so has the means to display adverts. For example, previous to the invention of widely used electronic media devices to spread information, advertising was done in the form of large billboards, pamphlets or even door-to-door knocking. As widely used electronic media devices such as the radio and television came into existence, advertising within these media did as well. Naturally, advertising also found a place within the Internet and the World Wide Web. The first company that offered commercial clients advertising space in the online world was Prodigy, founded in 1984. Prodigy was an online service that offered news, weather, games and other features over a network. At this stage, privacy was not a concern, however the Web was a stateless place. A stateless Web did not know who you are and what you had previously done even on a previous page. It made e-commerce very difficult as no shopping cart could be created. In 1994, to solve the statelessness of the Web, persistent client state HTTP cookies were introduced. A "Cookie" is a pass small data object passed between cooperating programs. The advent of the cookie technology also brought in companies that learned how to place their cookie on different domains. This allowed tracking users across websites. DoubleClick was the first company that used this technique to track users across websites in order to serve users ads tailored to their interests. As with many other forms of advertisements, the displaying of the ads was not an issue. Rather tracking users across websites is what developed into a privacy concern. The advent of cookies was the first method introduced to track users across websites. As

the public was made aware of this new technology, new tracking technologies emerged. We refer to companies that track users across different websites as *third-party trackers*.

1.1 Motivation, Problem Statement, and Summary of Contributions

Third-party trackers track users across different websites and build a *behavioral profile* on them. A behavioral profile is a profile that, based on a user's Web browsing habits, contains the interest categories of a user. Third-party trackers can also be referred to as *online profilers*. Information gathered on users can then be sold or used to directly target them with ads. In 2005, a data aggregation company called ChoicePoint sold 145,000 records to identity thieves [14]. The type of information gathered can be as detailed as menopause, getting pregnant, repairing bad credit and debt relief [2]. Furthermore, the types of ads targeted to users can undermine their personal autonomy. For example, it might be inferred from the behavioral profile of a user that he or she is an obsessive gambler who is about to quit gambling. Based on this information, an advertiser might offer gambling coupons to keep him or her addicted.

Different studies carried out with respect to targeted advertising and privacy shows that the majority of users are concerned with third-party tracking. We will see more on these studies in Chapter 2.

In our work, we will address the privacy concerns that arise in third-party tracking and *Online Behavioral Advertising* (OBA). OBA refers to the practice of tracking users across websites in order to infer user interests and preferences. These interests and preferences are then used for selecting ads to present to the user [1]. We investigate how tracking technologies have emerged and how trackers try to circumvent limitations put on them. We examine different solutions that have been proposed to solve privacy concerns

and argue that the current state of solutions does not address the concerns that have risen. We propose a novel idea that aims to achieve privacy in Web browsing by using obfuscation. To the best of our knowledge, no one has implemented a solution to achieve privacy in Web browsing using obfuscation. There has been work done to obfuscate the Web search queries of users (TrackMeNot), but not their browsing habits. We implemented our idea as a Firefox add-on, which is built on top of TrackMeNot. We called our solution ProfileMeNot. We argue that ProfileMeNot is a comprehensive solution that encompasses every tracking method that trackers employ to infer the behavioral interests of users. Via our experiment results, we demonstrate that it is possible to obfuscate the behavioral profile that trackers build on users. We further demonstrate that it is possible to control the amount of noise introduced. Our aim is not to block advertising or targeted advertising. Our goal is to create limited noise so that the business impact on advertisers is not so drastic that it will force them to turn their attention elsewhere, but for them to be encouraged to adapt an advertising model that respects privacy.

1.2 Organization

In order to better understand the problem and differentiate it from others, in Chapter 2 we examine targeted advertising and third-party tracking in greater detail. Furthermore in Chapter 2 we also define privacy, investigate user concerns, and describe what the law in North America says about privacy. To learn about and find a solution to third-party tracking, in Chapter 3 we investigate how third-party tracking came into existence and how it has evolved over time. To show the significance of our work, we present and examine different solutions that have been proposed in this field. In Chapter 5 we present

ProfileMeNot and the algorithms used in it. In Chapter 6 we explain our evaluation methodology and demonstrate our results. In Chapter 7, we conclude our work and discuss how our approach can be further developed.

2 Online Behavioral Advertising, Tracking and Privacy, a Background

In order to understand what the problem is and differentiate it from other fields, in section 2.1, we introduce and examine Targeted Advertising, Recommender Systems and Third-Party Tracking. Our goal is to address the privacy concerns that arise from online user profiling as a result of third-party tracking. As the primary goal of third-party tracking is to serve targeted advertising to the users, we also address the online-targeted advertising based on the behavior of the users. Additionally, most of the research done with regards to third-party tracking relates to online behavioral advertising. To better understand why privacy and profiling are concerns, in section 2.2 we will define privacy and online profiling, and examine different types of information that can be gathered as a result of user profiling. Section 2.2 will then move on to show statistics regarding user concerns and what the law in the United States and Canada says about privacy.

2.1 Targeted Advertising, Third-party Tracking and Recommender Systems

Targeted Advertising "is a type of advertising whereby advertisements are placed so as to reach consumers based on various traits such as demographics, psychographics, behavioral variables (such as product purchase history) and firmographic variables… or other second-order activities which serve as a proxy for these consumer traits" [4]. In the next few subsections, we will look at a few of the different types of targeted advertising that are related to the online world. We will further examine and differentiate between Targeted advertising and Recommender systems.

2.1.1 Contextual Advertising

The goal of advertising is to persuade an audience to take or continue to take a particular action. Market researchers and advertisers spend time researching how to effectively run their advertising campaigns to maximize their return on investment (ROI). In both online and other forms of media, such as newspapers, TV and cinema, ads can be targeted to meet the predicted audience of the content of the media. We classify this type of advertising as *Contextual advertising*. Contextual advertising can be formally defined as *displaying ads that have a direct correlation to the content of a webpage, newspaper, TV show or movie*. Contextual advertising is a form of Targeted advertising. For example, a visitor of www.nytimes.com might see a news related ad. This type of advertising does not require any type of user tracking and is targeted to an audience of a visited page or a purchased newspaper for example.

2.1.2 Online Behavioral Advertising, Analytics Services and Third-Party Tracking

The emergence of online advertising and tracking technologies has introduced a new form of targeted advertising: *Online Behavioral Advertising (OBA)*. OBA "refers to the practice of tracking users across Webs sites in order to infer user interests and preferences. These interests and preferences are then used for selecting ads to present to the user" [1].

The methods employed to track users across different domains are *tracking technologies*. The goal of some of the third parties is to track users across websites in order to infer their interests and build a behavioral profile on the user. They then use their interest to target the users with ads that the user is interested in, based on the profile that

they have built. Initially, it was online advertising networks that tracked this. However, as we will see in Chapter 3, the business has evolved and grown so that now, there are different players in the business. We refer to any entity that tracks users across different domains to infer some information about the user as a *third-party tracker*. In most cases, the end use of tracking users across different domains is targeted advertising.

With targeted advertising, users see ads that are targeted to their interests and behavior. Third-party trackers often claim that the data they gather is done anonymously. However, in reality, the data is pseudonymous, as trackers often assign either a cookie or some value to a user's browser, and try to build a behavioral profile on them and associate that profile with a unique identifier [40].

Ad-networks in the business of targeted behavioral advertising either buy information from other trackers or are trackers themselves. To target users with behavioral targeted ads, the party involved categorizes users into one or more *audience segments*. The segmentation is usually done based on the browsing behavior of users, but also can be done from their search terms [1]. The advertisers select the audience they would like to target their ads to. Currently, there does not exist a globally-accepted audience segmentation. Each company has their own audience segmentation break down. In Chapter 6, we will further explore Google's audience segmentation.

An example of third-party trackers is Web Analytic services. Web Analytic services track users across websites. Third-party analytics services provide tools for website owners to better understand visiting patterns, demographics, user agents, content views, etc. Most analytics services have adopted one of two business models: paid or free. The paid analytics services disclaim any right to access a client's analytics data except as

directed and take technical and business measures for it. The free method offers free analytics services in return for information. They monetize the information obtained from data collection by using it for ad targeting (e.g. QuantCast), market research (e.g. Google Analytics) etc. [2].

There is a difference between the nature of tracking in OBA and analytics. Analytics companies track users; however, the end goal of analytics is to help a particular website and is not based on a pseudonymous user.

2.1.3 Social Media Targeting

There are other types of targeted advertising that relate to the online world. A relatively new form of targeted advertising is *Social Media Targeting (SMT)*. SMT is "a form of targeted advertising that enables the customized placement of ads and communication activities on Web 2.0 platforms. SMT is a method of optimizing social media advertising by using profile data or delivering advertisements directly to individual users. Social media targeting refers to the process of matching social network users to target groups that have been specified by the advertiser" [63]. The difference between this type of advertising and OBA is that in Social Media Targeting (SMT), the users provide their information to the social networks voluntarily. This information includes age, gender, interests, and location. For example, the *Like* feature on Facebook lets Facebook know that you, for example, like a particular music genre or band. Although social media networks such as Facebook and MySpace might use technologies that OBA uses, social media networks have extra advantages. Users of social media networks are voluntarily logged in and also voluntarily and knowingly provide identifiable information. For example, a user will input his/her actual age, gender, birthday and some interests on a

social network. Another difference between SMT and OBA is that OBA infers a user's interest categories by tracking them across different websites. However, in SMT, the user, often on a single site, provides most of this information voluntarily. Given the above explanation, we believe that solving privacy in SMT is another problem that would require a different type of research. As we will see, ProfileMeNot, can be enhanced to create some type of noise in social networks in the case where the users are logged in. A more comprehensive solution has been proposed in [3].

2.1.4 Mobile Advertising and Location-based Services

Mobile advertising can also be a result of tracking and targeting users with personalized ads. However, Mobile Advertising has more elements to it than just browsing the Web. In mobile devices such as iPads, iPhones, and Android devices, in addition to having a browser, there are applications and a particular location tracker. Location can be tracked by GPS and an advertiser can serve localized ads. For example, user Alice is in downtown Ottawa, and because of her location she may see ads related to restaurants in downtown Ottawa. This type of advertising is called *Location-based services* and it aims to offer personalized mobile transactions for targeted individuals in specific locations at specific times, using the knowledge of the location of an object and/or individuals [5]. Localization can further be mixed with personalization and scheduling to serve much more targeted ads. For example, Alice is speculated to be a teenager who will finish school around 3:00 PM and is going to be in downtown Ottawa at around that time. She might be tired and hungry and hence will start seeing ads related to some nearby Pizza place. Mobile advertisements of this type could also be in a Short Message Service (SMS) format or a Multimedia Message Service (MMS) format.

With regards to applications, there are different ways to track mobile users and map them across different applications and also monitor their browsing habits. Device fingerprinting technologies are one way to track mobile users. "Fingerprinting technologies *anonymously* [sic] match a combination of attributes to a device to arrive at a high statistical probability that two events with a similar fingerprint are from the same device. For example, if someone from a specific device profile taps an ad and then a similar device profile registers an application install 90 seconds later, the probability that those two events were the same device is extremely high" [6]. The authors in [6] present a few other tracking technologies in mobile phones, including cookie tracking and Mac address. We will also examine fingerprinting with regards to OBA and third-party tracking in Chapter 3.

In summary, the key difference between mobile advertising and online behavioral advertising is that mobile advertising has added geographical location tracking. Mobile advertising can also time advertisements based on geographical locations, demographics and other factors. An example of such an ad would be advertising a fast food chain when Alice is about to leave school. Mobile OBA in the form that exists on desktop and laptop computers is only one way Mobile Advertisers can track and target users. ProfileMeNot could be used in conjunction with other tools such as MobiAd [7]. We, however, do not currently have the resources to develop an add-on and test ProfileMeNot on Mobile Advertisings. Another caveat in employing a solution such as ProfileMeNot in Mobile environments is bandwidth. While users might not mind sparing bandwidth on their desktop or laptop computers, they might be more conservative with regards to mobile phones as the amount of bandwidth available to users in this environment is limited.

2.1.5 Location-Based Advertising

In our work, we distinguish between *Location-Based Advertising* and *Location-Based Services*. We refer to Location-based ads as ads that are displayed to the user in an online environment, solely based on a wide geographical location (city or township etc.) That is not directly linked to a user's interest profile. An example of such an ad would be an advertisement of a grand library opening in a city. Marketers get to choose their audience by different interest segments. It could be that a localized ad is actually a targeted ad, but that is not always the case as user marketers might just target a particular area and not a particular interest group.

2.1.6 Recommender Systems

Recommendation systems seek to predict the preference or interest a user might have for a particular item (e.g. music, movie, book, laptop etc.). For example, based on a user's previous shopping habits, a recommendation system can suggest to a user which book they might be interested in, either based on the category of the book or based on a series of actions another user has taken. Recommender systems often try to predict an item that a user might be interested in which he or she might have not considered. One way of recommending items to users is through *collaborative filtering*. "Collaborative filtering methods are based on collecting and analyzing a large amount of information on users' behaviors, activities or preferences and predicting what users will like based on their similarity to other users" [8]. Examples of sites that utilize recommender systems are Netflix and Amazon. Recommender systems, while they might share their information with third-party trackers, gather their information differently from third-party trackers. Recommender systems are built on a site and per account basis and are not

based on users actions across multiple domains. The first party that the user interacts with is the one who recommends items to a user. Hence, this is not the problem that we are studying.

2.2 Privacy

In this section we examine definitions of privacy, profiling and personal information. We then move on to the philosophy and arguments regarding tracking and data collection. We then investigate the types of information that can be gathered based on tracking users across websites. We demonstrate an attack model that could reveal a user's interests. We then move on to looking at the extent to which users are concerned about online behavioral advertising and tracking. We conclude this section by investigating what law and regulation has to say about privacy with regards to online behavioral advertising and tracking in Canada and the US.

2.2.1 Privacy, Profiling and Personal Information

There is no consensus on the definitions of privacy, profiling and personal information. Different countries, different laws and different contexts define them differently. According to the Merriam-Webster online dictionary, privacy is defined as "1 a: the quality or state of being apart from company or observation, b: freedom from unauthorized intrusion <one's right to privacy>". In the context of online behavioral targeting, definition 1.b seems to be the right fit. Privacy has a perspective capacity. As different organizations and individuals see it as different things. For example, the way law in one country defines privacy, differs from another country, and differs from what organizations might consider privacy. This also differs from what a user might see as privacy.

In the online advertising world, the term profiling differs from *criminal profiling*. *Profiling* was first broadly used in connection with the training of crime profilers in the USA. The job of crime profilers was to determine a criminal's personality type by analyzing traces left at the scene of the crime [9]. In the context of the Web, profiling has also been defined as "a computerized method involving data mining from data warehouses, which makes it possible, or should make it possible, to place individuals, with a certain degree of probability, and hence with a certain induced error rate, in a particular category in order to take individual decisions relating to them" [9]. Another definition of profiling is "an automatic data processing technique that consists of applying a profile to an individual, namely in order to take decisions concerning him or her; or for analyzing or predicting personal preferences, behaviors and attitudes" [10].

As we saw in the previous section about tracking, profiling today, with the help of technology, not only analyzes databases to find data that is already known, but also attempts to "discover knowledge" that was not already known to be in the data [11]. From the definitions of profiling, we can conclude that behavioral advertisers are using profiling technologies and technics to target users [11].

TrackMeNot is a Firefox browser extension designed to achieve privacy in Web search by sending random fake queries to search engines [12]. The definition of privacy is subjective in general. Nissenbaum, one of the creators of TrackMeNot, in [13], understood privacy as "contextual integrity, or as flow of personal information that is consistent with context-specific informational norms, and we *operationalized* this as preventing access by search engines to the accurate record of your Web searches". They

define operationalization as developing concrete definitions of relevant values for the context of a given design project.

For our work, we build on top of the notion of profiling in [10] (taken from [11]) and define online profiling as: "*a computerized method involving data mining from data warehouses, which makes it possible, or should make it possible, to place individuals, with a certain degree of probability, and hence with a certain induced error rate, in a particular category in order to take individual decisions relating to them, decreasing their personal autonomy and liberty, repackaging and selling to other parties,"* and we consider online profiling by any other entity besides the user a *privacy violation*.

2.2.2 Philosophy and Arguments for Data Collection

Google's former CEO Eric Schmidt said, "If you have something that you don't want anyone to know, maybe you shouldn't have been doing it in the first place". Also, Facebook's founder Mark Zuckerberg said, "That social norm [of information sharing] is just something that has evolved over time" [14]. However, as we will see shortly, not only are Eric Schmidt's words inaccurate, it is not always 'things you do' that are being tracked.

There is the argument that says "if you don't want your data to be collected, simply do not use the facilities/tools that gather them". While this is a very simple and straightforward pronouncement to make, the social and personal costs of not utilizing such services are substantial and growing [14].

In the context of online advertising, the potential benefits of profiling for behavioral advertisers includes market segmentation, better analysis of risks and fraud, enhanced ability to adapt offers to meet demand, enhanced user experience, and the providing of

more relevant services and information [11]. We do acknowledge that many of the free content and services on the Web are the result of online advertising; we do not intend to kill the business model.

Proponents of third-party tracking usually argue that online tracking helps in providing better ads and hence increases revenue and free content on the Web.

On the other end of the spectrum, both behavioral profiling and the ads that are displayed to the user can undermine a user's personal autonomy and liberty. As an example, imagine that Alice is a smoker who is about to quit smoking. Her behavioral profile is matched with people who are about to quit smoking with a probability of 67%. With some other profile, it is predicted if she is offered free cigarettes with her online groceries and receives this news, then she will not quit with a probability of 80%. In this example, Alice's personal autonomy is being influenced by targeted advertisers. Profiling can also affect vulnerable groups. For example, an alcoholic will start seeing advertisements related to cheap alcoholic products. Consumer profiling could potentially target people with eating disorders, advertising pills to help lose weight [11]. Other vulnerable groups could be profiled under people who have an unhealthy diet, and later be denied health insurance.

The global spending on digital advertising in 2012 was around $102 billion. Of this, 39% is the share of North America. Currently, around 1 out of every 5 dollars spent on advertising is spent on digital adverts. It is projected that online ads sales will reach $118.4 billion in 2013 [15]. Online adverts are the main reason many of the online services and information remains free. One of the strongest arguments against profiling users and targeting them with personalized ads is a study which shows that as of 2009,

only around 4% (less than one billion dollars of US online advertising expenditure) of online ads were targeted ads. This figure is projected to be around 7% by 2014 in the United States [16].

2.2.3 Type of Information Gathered, Sold and Attack Model

It is hard to say what type and to what extent the information of users is gathered. This is due to the fact that advertising networks keep this data private. However, some estimates can be made. There have also been some leaks, which give us an idea in this regard. The websites an individual visits are inextricably linked to his or her personal information. Page visits can reveal a person's location, interests, purchases, employment status, sexual orientation, financial challenges, medical conditions, religion, political view, gender, age, habits and much more [2].

In mid-2011, Jonathan Mayer discovered that an advertising network, Epic Marketplace, had publicly exposed its interest segment data. Among the user segmentations was menopause, getting pregnant, repairing bad credit and debt relief. They also found that the dating site OkCupid was sending data to provider Lotame about how often a user drinks, smokes, and does drugs [2].

Over the Internet, every click and page view maybe be logged and analyzed. This data can be sold to third parties and also acquired by a variety of means [14]. In 2005, ChoicePoint, a data aggregation company, sold 145,000 records to identity thieves. ChoicePoint is just one example. This shows that we are not always aware of how our information is repackaged and sold.

Castelluccia et al. [17] demonstrate that targeted ads expose users' private data not only to ad providers but also to any entity that has access to users' ads. They find that an

adversary that has access to the Google ads displayed can infer a user's interests with an accuracy of more than 79% and reconstruct as much as 58% of a Google Ads profile. Their work is, to the best of their knowledge and ours, the first effort that quantifies information leakage through ads served in targeted advertising. Among Google's ads interest segmentation is *People & Society→ Family & Relationships → Marriage*. This information from a reconstructed profile could indicate that a person is about to propose to someone or is about to get married, which is information that some people would like to keep private. Another interest category that a user might find private or sensitive is information regarding adopting children. If an attacker can reconstruct *Family & Relationships → Family→ Parenting → Adoption* from the list, then this might reveal that a person is about to adopt a child.

2.2.4 Statistics on User Concerns

Some studies have been done to learn about user's concerns regarding online behavioral advertising. Joseph Turow et al. [18] in a representative survey of Americans found that 66% of adult Americans do not want marketers to tailor advertisements to their interests. Furthermore, they were asked if they are ok with ads being tailored to them based on three tracking activities; "the website you are visiting", "other websites you have visited" and "what you do offline-for example, in stores", and between 73% and 86% said they don't want to be tracked. When they were told that they will remain anonymous, still 68% said that they "definitely would not allow it", and 19% would "probably" not allow it.

A December 2010 poll done by Gallup asked the question: "As you may know, website advertisers are currently able to match advertisements to your specific interests. They do

this by collecting data that shows what websites you have visited. Do you think advertisers should or should not be allowed to do this?" 67% of respondents said no, 30% said yes they are should be allowed and 3% had no opinion. [19].

In a July 2011 TRUSTe survey [20], the following information was obtained:

- 35% said that they are aware of the term OBA (Online Behavioral Advertising)
- 70% said they were aware of the OBA concept
- 43% have a negative connotation with the term
- 54% do not like the concept of OBA
- 37% have felt uncomfortable with a targeted online advertisement based on their browsing behavior or personal information

2.2.5 Privacy and the Law

Law and technology both have the power to organize and impose order on society. As Helen Nissenbaum puts it, "law and technology can both systematically impede others; both have capacities to enable, constrain, allow and prevent" [28]. In some cases, technology enforces the law and makes breaking the law very expensive. For example, on a two-way highway, it is illegal for a driver to make a U-Turn at some sections of the road. To enforce this, there are road dividers placed on the street. Unfortunately, in the context of online behavioral advertising, not only is there not such an enforcing technology, but also regulations are subject to interpretation as they are not explicitly tailored for OBA.

It's also important to remember that different countries have different laws. For the purpose of our work, we will look at the laws and regulations in two places; Canada and the USA.

In Canada, the Personal Information Protection and Electronic Documents Act (PIPEDA) is the legislation that privacy of data collection from third-party companies falls under. PIPEDA is for the private sector that operates within Canadian jurisdiction. Some Canadian provinces have enacted similar laws that supersede PIPEDA within their jurisdiction. This Act generally applies to collection and use or disclosure of information. PIPEDA also applies to extraterritorial organizations that engage in trans-border flow of personal information, such as Google. PIPEDA has 10 principles, the third principle states "knowledge and consent of the individual are required for the collection, use, or disclosure of personal information, except where inappropriate". The 'inappropriate' section is allowed only under certain circumstances, where collection is clearly in the interests of the individual and the personal information cannot be otherwise accessed. With regards to online advertising, the privacy commissioner of Canada is of the view that 'reasonable people' do not expect that organizations use their personal information in the online targeted advertisers do [21]. The enforcement of PIPEDA relies on an ombudsmen model [22]. Given the nature of the model, PIPEDA does not create automatic right to sue for violations; complaints are taken to the Office of the Privacy Commissioner of Canada. The commissioner, based on the complaint, has to create a report. However, the report is not binding and none of the parties are required to comply. However, PIPEDA gives the complainant the right to apply to the Federal Court of Canada for a hearing, and the court can take further actions [23].

In the US, privacy laws are not as explicit as they are in Canada. The Federal Trade Commission (FTC) is the body in charge of privacy issues and the private sector. The FTC can only prevent business practices that are either "unfair" or "deceptive". Privacy,

third-party tracking and OBA fall under deceptive business practices. "The FTC almost always settles a company's first violation with a consent order and slight (if any) payment. A subsequent violation of a consent order can result in significant monetary penalties" [2]. No law or FTC regulatory action currently requires behavioral advertisers to give consumers access to their personal information or profiles. "No federal statute of general application comprehensively regulates businesses to protect consumers' privacy or data protection and no federal laws have been yet adopted to specifically regulate behavioral advertising practices or automated profiling" [11]. Since late 2010, the FTC has called for a *Do Not Track* (DNT) consumer choice mechanism [2]. As we will see, DNT is universal, usable and persistent. The FTC can take action against unfair or deceptive practices in commerce; however, it has not taken a position on whether new law or rulemaking is necessary for DNT [24].

On the flip side of DNT, Rebecca Balebako et al. [25] found that DNT headers were not particularly useful. In April of 2011, US Senators John McCain and John Kerry introduced the Commercial Privacy Bill of Rights. This bill, although an improvement, explicitly mentions that "… *the bill would require robust and clear notice to an individual of his or her ability to opt-out of the collection of information for the purpose of transferring it to third parties for behavioral advertising"* [26]. The bill however, does not entail third-party trackers who could potentially sell their information to online advertising networks. Another of the bill's defects is that it emphasizes the regulation of information use and sharing, not on the collection of data. Under this bill, it is easier for users to opt-out; however, as we will see in Chapter 3, there are many third-party trackers and doing so is very tedious and often many of the smaller tracking companies are not

known. The bill fails to mention DNT, a universal opt-out mechanism that would not require users opting out of every tracking company one by one. The bill also doesn't give users a private right of action, meaning that they cannot sue companies if the provisions of the bill are violated [27].

In a nutshell, law and regulation have the potential to impact online privacy and data collection. However current provisions, especially in the United States are not enough to get round all the loopholes and enable a global DNT mechanism. There are also no technologies, as in traffic laws, that enforce the current laws. We can conclude that at least until law catches up with user concerns, such as an enforced and audited global opt-out mechanism, we have to look for technological solutions to solve the issue. As former US Senator and current Secretary of State John Kerry puts it [27]:

"*Companies can harvest our personal information online and keep it for as long as they like. They can sell it without asking permission or even letting you know that they're selling your own information. You shouldn't have to be a computer genius in order to be able to opt-out of information sharing.*"

2.3 Conclusion

In this chapter, we examined different types of advertising and defined what OBA and third-party tracking are. We concluded that OBA is the predominant reason that profilers share and sell users' information across different websites. Analytics companies also track users; however, the end goal of analytics is to help a particular website that is not based on a pseudonymous user. However, the problem arises when analytics companies gather information from different websites, track users and either sell the information to ad networks or use it themselves for targeted advertising. The same argument applies to

any other tracking company that tracks users and sells their information. In the section after, we defined online profiling and investigated the type of information that has been and can be gathered from tracking and profiling users. We also saw that the majority of Web users are concerned about being tracked online and having a behavioral profile built on them. We then investigated the current laws and regulations in place in North America and noted that the current state of the law is not enough.

3 Online Advertising and Tracking Technologies

In order to find a solution to the privacy concerns raised in online third-party tracking and OBA, it is important to understand how the technology in this field has developed and where it stands now. The information in this chapter also helps in better understanding the strengths and weaknesses of existing technological solutions that have been proposed and/or implemented to solve privacy concerns of users. In this chapter, we investigate how online advertising came into existence. We then move on to how online behavioral advertising started and how tracking technologies emerged. We conclude by examining how tracking technologies have evolved over time.

3.1 Online Advertising

The first company that offered commercial clients advertising space in the online world was Prodigy, a company founded in 1984. Prodigy was an online service that offered news, weather, games and other features over a network. Prodigy started online advertising by promoting products from the superstore Sears. At the time, IBM and Sears owned Prodigy. Prodigy was the first online company to offer its users a friendly graphical user interface. It used its graphical capability for advertising; however, due partly to the fact that graphical technologies back then could not offer a realistic picture of products, it was difficult for online merchants to market their products. Hence Prodigy did not make as much money as it initially thought it would from advertising.

In May 1993, the Global Network Navigator (GNN) was founded. GNN was the first commercial Web publication and the first website to offer click-able ads. These ads were later named banner ads. In this new model of advertising, ads are placed on Web pages and users who click on them are redirected to the advertisers' website or product page.

The first company to serve such ads in large quantity was HotWired. As a matter of fact, they were also the first to measure the effectiveness of online advertising. In the next sub-section, we will look at how online advertising gave birth to OBA and third-party tracking, and their examining its history.

3.1.1 Online Behavioral Advertising, A History

The history of Online Behavioral Targeted Advertising (OBA) goes hand-in-hand with the history of cookies. This history is comprehensively explained in [28]. In this section, we will mostly summarize work done in [28].

Before the invention of cookies, the Internet was a stateless place. A stateless Web was devoid of memory; it did not know who you are, what you did, your previous purchases, purchase habits etc. This made commerce on the Web very difficult. The Enterprise Server Division of Netscape Communication Corporation had a contract for a new shopping cart application for online stores. “A Shopping cart would allow a website to keep track of multiple items that a user requested”. At the time, state information was stored in the URL. This method was not very effective since malicious users could tamper with the URL and hence led to the idea that state data must be stored somewhere other than the URL. Eventually, Lou Montulli and John Giannandrea came up with the persistent client state HTTP cookies. A “Cookie” is a small data object passed between cooperating programs. Eventually, the first use of cookies was by Netscape to determine if visitors to its website are first time visitors or not.

The cookie technology was quickly integrated into Netscape's Web browser in 1994. Netscape integrated the cookie feature by default without notifying or asking the consent

of users. There was also no documentation to tell them what cookies were and what privacy risks they entailed.

3.1.2 Cookies as a Privacy Risk

On February 12th, 1996, the Financial Times published a story that explained what cookies are to the public for the first time. Following the article, cookies became a top-priority in Internet privacy.

The Internet Engineering Task Force (IETF) initially proposed standards that were based on a technology different from cookies; however, given the ubiquity of the Netscape cookie model, it switched to Netscape's model. Their goal at this stage was to develop a precise standard for cookies, but they inevitably ran into privacy and security problems. The most serious problem they encountered was *third-party cookies*. The original intent of the Netscape team was that cookies made by a particular website be readable by only that website. In 1997, cookies were introduced as a standard in Request for Comments (RFC) 2109 to the Network Working Group (NWG) of IETF. In it, Montulli and his colleague David Kristol made their intention clear when they wrote in the RFC, "The intent is to restrict cookies to one, or a closely related set of hosts. ... We consider it acceptable for hosts host1.foo.com and host2.foo.com to share cookies, but not a.com and b.com" [13]. For example if carleton.ca placed a cookie on a computer, waterloo.ca would not be able to read or modify this cookie. However, the advertising industry found a way to circumvent the restrictions put on cookie exchanges. DoubleClick attached its own cookies into people's browsers by attaching them to ad images incorporated into websites that were visited. This created a loophole that let third parties read and write cookies and allowed companies like DoubleClick to track users

across websites. For example, DoubleClick put ads on Nytimes.com, and hence it could store a cookie on a browser. Once the user goes to abc.com, where DoubleClick also places ads, DoubleClick can then view and modify its previous cookies and then place an ad on another site based on the interests it has inferred from a user. Online advertising companies, now a new and powerful force, opposed the RFC 2109 standard that restricted cookies to one or closely related hosts and instead supported RFC 2965, which allowed the placement of third-party cookies. Ultimately RFC 2965 was victorious [13]. The RFC 2965 even let browsers allow third-party cookies by default.

The discussions regarding cookies in the media led to some action by the Federal Trade Commission (FTC) of the Unites States. The FTC is an independent agency of the US government, which aims to promote consumer protection and eliminate and prevent anti-competitive business practices [30]. The US government did not ban third-party cookies; however, it did push browsers to provide cookie management tools and provide improved documentation on cookies [28].

In February 2000, the Electronic Privacy Information Center was the first to file a formal complaint against DoubleClick to the Federal Trade Commission among other suits that followed. On March 29, 2002, a federal US court granted them preliminary approval to settle all charges related to privacy of web surfers. Among the provisions in the preliminary settlement was that DoubleClick would expire its cookies after five years. This was not favorable to Marc Rotenberg, executive director for the Electronic Privacy Information Center, as he noted that most users change their computers every two to three years. Another of Marc's concerns was that the agreement didn't include any access to profiles created from users [29].

On May 21st, the US district court of southern New York granted the final approval of the settlement. The settlement required DoubleClick to clearly, in "easy-to-read" sentences, notify users in the privacy policy about its online advertising service and use of cookies. The settlement also required DoubleClick to perform a public information campaign via 300 million banner ads to educate the public. In addition, the settlement also required for DoubleClick to only collect personally identifiable information linked to Web search with the users' permission [31].

Eventually in March 2008, Google acquired DoubleClick for 3.1$ billion. On 11th of March 2009, Google AdSense announced that it will "provide interest-based advertising across AdSense publisher sites." In their announcement, they explicitly mention that they will "recognize the types of web pages" that users visit across their network [20]. However, Google does allow users to see the profile that Google has built on them and lets them modify it. Interestingly, it also allows users to opt-out. The page is called *ads preferences* and the information is gathered through a cookie associated with a browser. A sample Google Ads Preferences page is shown in Figure 1.

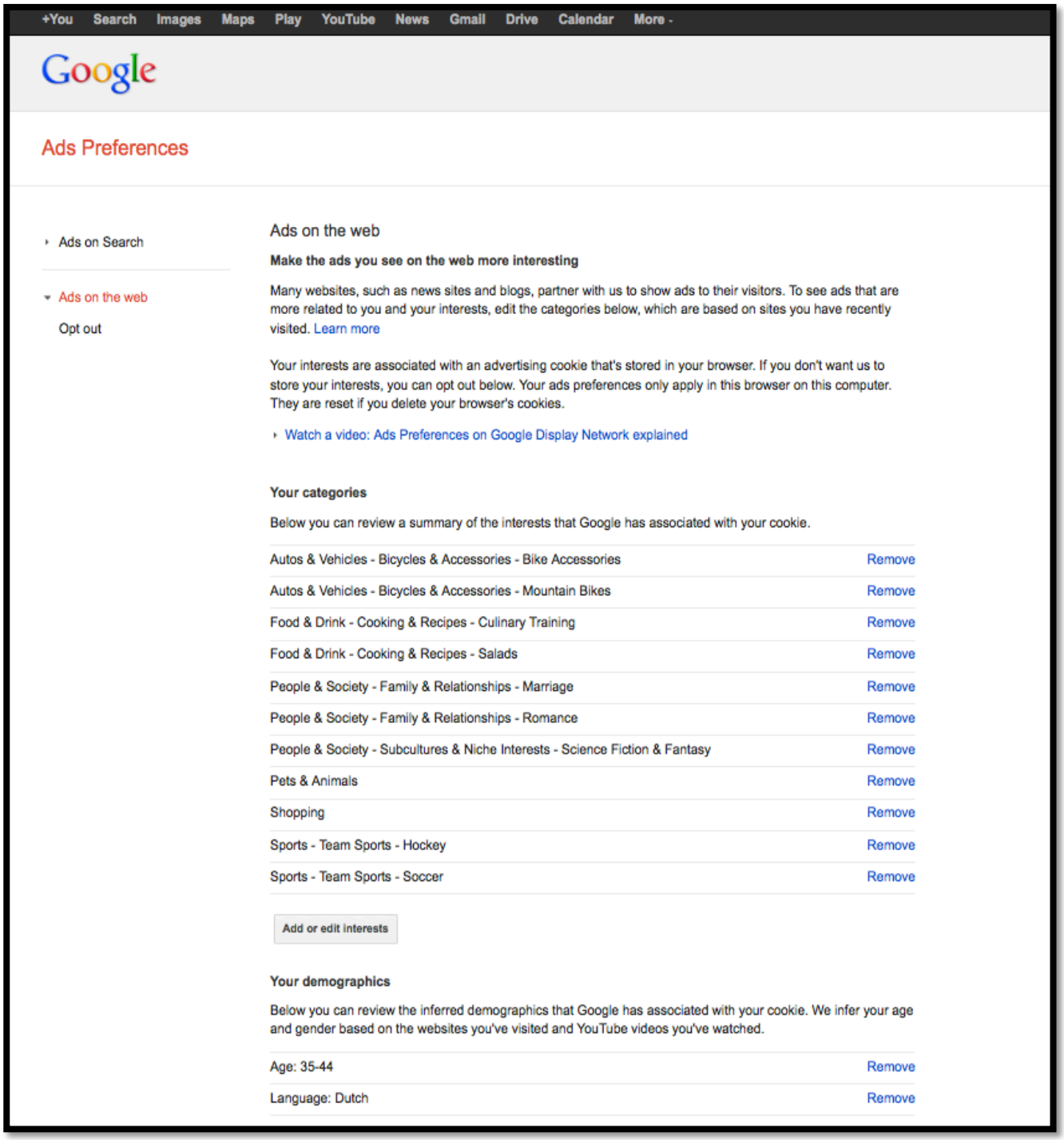


Figure 1: Sample Google ads preferences

Google and DoubleClick are not the only companies that use tracking technologies for the purpose of serving better ads.

3.1.3 Development of the Online Advertising Business Model

Since the emergence of the cookie, the online advertising industry has turned into a mature business model. In a nutshell, there are three major players in the online advertising industry [1]: 1) Ad-Networks; 2) Advertisers; 3) Publishers.

Advertising Networks (Ad-Networks): By the late 1990s, growth in advertiser demand and ad slot supply made it impractical for advertisers and publishers to deal with each other directly. A new party emerged in the online advertising world, the advertising network [2]. The advertising network is a party that collects ads (and payment) from advertisers and places them on publisher pages (along with paying the publisher). Examples of ad-networks include Google, Yahoo!, MSN, AOL, and AdBrite. Often, these are the companies that track users across the Web.

Advertiser: a party that has an online ad it wants to embed in web pages across the Web. The advertiser is willing to pay for this service.

Publisher: A party who owns a webpage (or website) and is willing to place ads from others on its pages. The publisher expects to be paid for this service.

The cores of the online advertising business are the three aforementioned parties. However, as the business has grown in size, the services have broken down into more detailed tasks. In the mid 2000's, ad networks alone could not fill all the ad slots available for publishers. Hence, publishers began finding ways to monetize the remaining slots they had. Ad exchanges offered this service in real time, taking bids from different advertisers via different ad networks. Furthermore, the industry expanded more and

several intermediary businesses appeared. The most relevant to our work is the creation of data providers. Data providers sell ad-targeting data to advertisers in real-time. Data providers often base their targeting criteria from tracking (e.g. Quantcast) or information purchased from publishers (e.g. BlueKai).

There are also other businesses that may benefit from the online profiling of users, such as analytics services, content providers, frontend services, hosting platforms and market trends.

Ad-networks have also developed different methods to bill advertisers. Advertisers either bill per impression or per click. Ad per impression, as the name suggests, bills advertisers per the number of times an ad is viewed. Within this scheme, the advertiser can pay more to specify the target audience they would like to advertise to. Ad per impression is usually sold in per thousand impressions, known as *cost per mille* (CPM). This is done in order to facilitate billing. In the ad per-click model, the advertisers pay per each click on an ad. Cost per impression can be as low as 10 cents per 1,000 impressions.

3.1.4 Third-Party Content in First Parties

A question that might arise in targeted advertising is how do first parties let third parties embed content on their site. As we saw in section 2.2.1.2, publisher websites let advertisers place an ad on their site. While doing so, they permit the third-party company to set cookies and other information on the user's computer. Furthermore, it is not only through ads that third-party tracking companies such as ad networks are able to track users. First parties also allow third-party tracking companies to embed on their pages a link to the ad-network or other tracking companies in one of the following forms:

- *iFrame*: An iFrame is an inline frame used to embed another document within the current HTML document.
- An Image
- *Web bug* (web beacon, tracking bug): "A web bug is an object that is embedded in a web page or email and is usually invisible to the user but allows checking whether a user has viewed the page or email" [32]. A web bug could be a 1*1 pixel transparent GIF or PNG tag.

The next question that might arise is how do third-party companies learn about the first party company that they have been embedded on, as they do not have direct access to the first party site. This turns out to be a straightforward process. As third parties are allowed to embed content on first party websites, they are made aware of the URL of the first party header through an HTTP referrer. The HTTP referrer header is an HTTP header field that identifies the address of the webpage that linked to the resource being requested [32]. Some third-party content reports a first-party page's URL as a parameter in a request. Using the LiveHTTPHeaders add-on on, we inspected some of the HTTP requests that were being sent and received. Figure 2 is an example of third-party content on www.nytimes.com getting aware of which first party URL is referring it. Please note that we omitted and shortened some of the fields.

http://b.scorecardresearch.com/p?cs_**iframe** …
GET /p?cs_**iframe** … Fwww.nytimes.com%2F&c9= HTTP/1.1
Host: b.scorecardresearch.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; rv:19.0) Gecko/20100101 Firefox/19.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referrer: http://www.nytimes.com/

Figure 2: Example of an iFrame and how it learns about its referrer

3.2 Emerging Tracking Technologies

After the use of cookies as a means to track users, other tracking technologies have emerged. Users are more aware of cookies and even browsers give users the option of blocking them or blocking third-party cookies. Although these means are effective in blocking some forms of tracking, blocking cookies does not block other types of tracking. In this sub-section, we will look at some new tracking technologies that have emerged.

Stateful tracking are tracking technologies that store a state. Stateful Tracking technologies, except for HTTP cookies, are also known as *Supercookies*. "A website can encode a globally unique pseudonymous device identifier into any stateful Web technology so long as it persists at least $log_2 n$ bits, where n is the number of Internet-connected devices (presently roughly 5 billion, requiring 33 bits)" [2].

We will dedicate the next few subsections to different supercookies. We then dedicate one subsection to fingerprinting and conclude with a table summarizing different tracking technologies.

3.2.1 Flash Cookies

A new method to track users is through *Local Shared Objects* or *LSOs*, also known as flash cookies. The reason they are called flash cookies is because they are pieces of data that are stored on a user's computer through all versions of Adobe's flash player. Flash cookies can store up to 100 KB of data, as opposed to 4KB HTTP cookies and unlike HTTP cookies, do not have expiration dates by default. Flash cookies are not known to the public and are stored in a different location than HTTP cookies. On top of that, browsers don't have a standardized way of giving the user control over flash cookies. Prior to 2010, private browsing mode did not affect flash cookies. Flash cookies are also stored at the same location for all browsers; hence websites can see flash cookies that were set by other browsers. Flash cookies of different websites are also easily read by other websites.

A study conducted by Ashkan Soltani et al. [33] in 2009 on the most popular 100 websites found that more than 50% of the sites were using flash cookies to store information about the user. Some of their key findings are the following:

They found that 54 of the top 100 websites used flash cookies: “A total of 157 Flash shared objects files yielding a total of 281 individual Flash cookies”. In contrast, 98 of the websites stored a total of 3,602 HTTP cookies. They also found that 31 sites had at least one overlap between an HTTP cookie and Flash cookie. Most of the matching cookies were from third-party trackers. They found 37 matching HTTP and Flash cookies from the following advertisers: ClearSpring (8), Iesnare (1), InterClick (4), ScanScount (2), SpecificClick (14), QuantCast (6), VideoEgg (1), and Vizu (1).

The issue arose regarding whether or not websites and tracking networks are using Flash cookies to accomplish redundant unique user tracking. This would allow websites to backup HTTP cookies. They found that several websites would *respawn* a deleted HTTP cookie. An example of such a site was Hulu.com. A QuantCast Flash cookie respawned a deleted QuantCast HTTP cookie. Even more surprising was the respawning of HTTP cookies across domains. A third-party ClearSpring Flash cookie respawned a matching Answers.com HTTP cookie.

Some of these websites that used Flash cookies belong to Network Advertising Initiative (NAI), a US-based self-regulatory project hosted by a public relations firm. NAI members must offer an opt-out cookie from OBA. They found that Flash cookies were still used when the NAI opt-out cookie was installed for QuantCast on browsers. QuantCast cookies were re-spawned after QuantCast HTTP cookies were deleted; however the opt-out cookies were not re-spawned.

Of the 100 websites, only 4 of them mentioned the use of Flash files as a tracking system in their privacy policy.

To summarize their study in 2009, Flash cookies can be used to circumvent user deletion or blocking of HTTP cookies. However, their findings were not fruitless in the industry. Right after the publication of their findings, QuantCast contacted Soltani and Hooflange and they changed the respawning behavior.

In [34] Mcdonald et al. quote a W3C document on mobile Web use:

“Cookies may play an essential role in application design. However, since they may be lost, applications should be prepared to recover the cookie-based information when

necessary. If possible, the recovery should use automated means, so the user does not have to re-enter information."

Furthermore, Flash cookies can be used to uniquely track users even if they do not respawn. Even if users delete HTTP cookies, they still might be uniquely tracked via Flash cookies from first or third parties [34]. Even though in such a case the cookies are not being respawned, it is functionally the same as respawning. However, not all Flash cookies are used for tracking purposes. They are also used for the purpose they were created for- to save audio and video settings.

In January 2010, Adobe made Flash Player compatible with "Private Browsing Modes" in major Web browsers. There have also been browser extensions built to manage Flash Cookies [35]. The Federal Trade Commission also did not miss Flash cookies. In its staff report on privacy in December 2010, the problem was acknowledged:

"*…Consumers are not likely to be aware of the technical limitations of existing control mechanisms. For example, they may believe they have opted out of tracking if they block third-party cookies on their browsers; yet they may still be tracked through Flash Cookies or other mechanisms…"*

In 2010, the NAI also posted its policy on "Flash cookies" and "similar technologies".

"*NAI members have confirmed that they are not using Flash cookies for online behavioral advertising (OBA). The NAI in 2010 took the position that its members should not use locally-shared objects (LSOs)* like Flash cookies for OBA, Ad Delivery & Reporting, and/or Multi-Site Advertising, until such time as Web browser tools allow for the same level of transparency and control as is available today for standard HTTP cookies. …*" [36].

In July of 2011, Ayenson et al. did another study on flash cookies [35] based on the work done in 2009 [33]. In this study, they also studied HTML5 cookies and ETags. They crawled the most popular 100 US websites based on QuantCast.com's ranking as of July 2011. We will mention their findings in a separate sub-section. The key findings with regards to Flash cookies were the following:

There are a total of 5,675 HTTP cookies stored from the top 100 US websites. Twenty sites placed 100 cookies or more. Of the 5,675 cookies, 4,915 of the cookies were from third parties. Google and DoubleClick combined had cookies on 97 of the top 100 sites. Among the 100 sites, scorecardresearch.com with cookies present in 61 sites and atdmt.com with cookies present in 56 sites were the next prominent third-party trackers. Based on an analysis we did ourselves (the authors of this thesis), we found that tools such as AdBlock do not currently block cookies from scorecardresearch.com.

They found 100 Flash cookies on the top 100 sites in 37 sites. Two sites (foxnew.com and hulu.com) had shared values between Flash cookies and HTTP cookies.

In January of 2011, before the work done by Ayerson et al., McDonald and Cranor carried out another work in the literature of Flash cookies. They conducted a survey on the use of Adobe Flash cookies to respawn HTTP Cookies [34]. Their work is similar to that of [35]. They found that that only two of the top 100 sites respawned HTTP cookies from Flash cookies, and they stopped respawning after their work. On top of the 100 top websites, they checked for respawning of HTTP cookies from Flash cookies from 500 randomly selected websites. They found that none of the 500 randomly selected websites respawned HTTP cookies from Flash cookies. Among the top 100 websites, they found that 20 stored Flash cookies and 9 used cookies to store unique identifiers. In the 500

random sites, Flash cookies were set in 41, and 17 used their Flash cookies to store unique identifiers. Table 1, taken from [35] summarizes the three works cited in this section. Note that their methods of investigating the sites were different.

	Soltani 2009	McDonald 2011	Ayenson Wambach et al. 2011
Number of sites with Flash cookies (top 100 sites)	54	20	37
Total Number of Flash cookies (top 100 sites)	281	Not reported	100
Sites with respawning (top 100 sites)	6	2	2
Number of websites with HTTP Cookies (top 100 sites)	98	98	100
Total HTTP Cookies set (top 100 sites)	3,602	Not reported	5,675
Sites with shared Flash/HTTP values on top	31	Not reported	2
Total shared Flash/HTTP values on top 100	41	8	2
Sample	Top 100 websites and six government sites	Top 100 websites and 600 random sites	Top 100 websites
Method	Visited homepage and then made 10 clicks on the same domain	Visited homepage multiple times	Visited homepage and then made 10 clicks on the same domain

Table 1: Comparison of findings in different studies with respect to Flash cookies [35]

The conclusion that can be drawn from Flash cookies over the three works is that the use of Flash cookies is on the decline. However, other technologies exist and will continue to be developed in the future. As McDonald and Cranor mention, rather than

focusing on individual technologies, the larger picture has to be seen and solved, or else we will fall into an arms race with advertisers changing the technologies they use to track users.

3.2.2 HTML5 Storage

HTML5 storage has been developed to overcome the limitations of cookies. HTML5 storage is a client-side storage that is session based. As per W3C Recommendations, this allows storage of up to 5 MB of data whereas cookies only allow you to store 4KB of data. The data is stored in key/value pairs. HTML5 cookies allow for storing two types of objects: *localStorage* and *sessionStorage*. As its name suggests, localStorage stores the data objects with no expiration date and remains after the user has closed their browser. Unlike HTTP cookies that have a default expiry date, HTML5 localStorage is only deleted when a user or website deletes them. On the other hand, sessionStorage is deleted once the user closes their browser. For more information, we refer the readers to [37].

As with HTTP and Flash cookies, HTML5 cookies can be used to track users across different domains. HTML5 storage is more persistent than HTTP cookies, as they do not have a default expiry date (in the localStorage version) and can store up to 5MB of data by default, as compared to 4KB by HTTP cookies.

In the same study we saw above [33], they also studied HTML5 Web storage. They found that among the top 100 websites, seventeen were using HTML5 storage. The 17 sites had a total of 60 key/value pairs. They found matching values among HTML5 local storage and HTTP cookies on a few sites. “Twitter.com, tmz.com, squidoo.com, nytimes.com, hulu.com, foxnews.com, and cnn.com had such matching values; in most of

these cases, the matching value was with a third-party service, such as meebo.com, kissanalytics.com, and polldaddy.com."

As we see, another evolving tracking technology is HTML5. Table 2, taken from [33], summarizes the tracking technologies we have seen so far.

	HTTP Cookies	Flash Cookies	HTML5 Storage
Storage	4KB	100KB	5Mb by default
Expiration	Session by default	Permanent by default	Permanent by default
Location	In SQL file (Firefox)	Stored outside the browser	In SQL file (Firefox)
Access	Only by browser	Multiple browsers	Only by browser

Table 2: Comparison of tracking technologies [33]

The story of tracking technologies does not end here. Ayerson et al., in their research found that first party HTTP and HTML5 cookies respawned on hulu.com through a service hosted at kissmetrics.com. However, this respawning wasn't related to Flash cookies, but to utilize the cache to retrieve values. They were using *ETags*, which we will look at in the next section.

3.2.3 HTTP ETag

"An ETag or entity tag is part of HTTP, the protocol for the World Wide Web. It is one of the several mechanisms that HTTP provides for Web cache validation, and which allows a client to make conditional requests. This allows caches to be more efficient, and saves bandwidth, as the Web server does not need to send a full response if the content has not changed" [38]. In 2003, Dean Gaudet pointed to unique user tracking through using "ETags" [35]. ETags are a form of fingerprint that identify if the content of a URL has changed or not. They are stored as part of a browser's cache. The methods deployed to generate ETags are not specified by HTTP specifications. Typically, when a URL is retrieved, the Web server returns an ETag field in its HTTP response. The user's browser

can choose to cache the ETag field. If the user then wants to retrieve the same URL again, the user's browser will send its cached ETag value. If the value of the URLs current ETag and the one the user's browser sent match, then no new content is sent [38].

Ayerson et al. note that ETag tracking and the respawning tracking technique generates unique tracking values even where the consumer blocks HTTP, Flash, and HTML5 cookies. ETags can even track the users in private browsing mode. In order to block tracking from ETags, users would have to clear their cache between each website visit.

3.2.4 Fingerprinting Browsers

Technically advanced privacy conscious users, as far as we have seen in this thesis, have two challenges to overcome. Firstly they have to find a way to prevent third-party trackers from using HTTP cookies to track them. Note that blocking HTTP cookies will have a drastic impact on a user's experience while browsing. The second challenge they face is to learn about different supercookies and find ways to disable them. Most users, even the technically advanced ones, will most likely not completely overcome the second challenge. After the two, there comes a third challenge; fingerprinting. In the remainder of this subsection, we will summarize the work done by Peter Eckersley for the Electronic Frontier Foundation [39].

In [39], the authors implemented a browser-fingerprinting algorithm by grouping 8 different strings from commonly and less-commonly known characteristics that browsers make available on their websites. The fingerprint generated is a sort of concatenation of these 8 strings. These eight strings are summarized in the following table 3.

Variable	Source	Remarks
User Agent	Transmitted by HTTP, logged by server	Contains Browser micro-version, OS version, language, toolbars and sometimes other info.
HTTP ACCEPT headers	Transmitted by HTTP, logged by server	
Cookies enabled?	Inferred by HTTP, logged by server	
Screen resolution	JavasScript AJAX post	
Time zone	JavaScript AJAX post	
Browser plugins, plugin versions and MIME types	JavaScript AJAX post	Sorted before collection. Microsoft Internet Explorer offers no way to enumerate plugins; they used the PluginDetect JavaScript library to check for 8 common plugins on that platform, plus extra code to estimate the Adobe Acrobat Reader version.
System fonts	Flash applet or Java applet, collected by JavaScript/AJAX	Not Sorted
Partial supercookie test	JavaScript AJAX post	They did not implement tests for Flash cookies, Silverlight cookies*, HTML5 databases, or DOM globalStorage

Table 3: Browser fingerprint elements [39]
*: Silverlight cookies, HTML5 databases and DOM (Document Object Model) global storage are other methods websites can store data on a browser.

To generate fingerprints, they created a link at panopticlick.eff.org and for users who clicked on 'test me', within the site, they generated fingerprints. They found 83.6% percent of the browsers observed on Panopticlick were unique. Of the users who had Java or Adobe Flash installed, this value was 94.2%. Of the remaining 5.8% of users that had Java or Adobe Flash installed, 4.8% of the fingerprints were seen only twice. To be able to effectively calculate the number of times that a fingerprint was generated by more than one browser, each user that clicked on 'test me' had the fingerprint of their browser recorded, on top of a three month HTTP cookie, an HMAC (hash message authentication

code) of the IP address and an HMAC of the IP address with the least significant octet erased. The key of the HMAC of the IP address was later discarded. Many of the users heard about this site through websites like Slashdot, BoingBoing, Lifehacker, Ars Technica, io9 and through social media platforms like Twitter, Facebook, Digg and Reddit. Their sample of testers, as they describe themselves, is a biased sample geared towards the technically savvy and privacy-conscious users.

The fingerprints are subject to change, as a user might update plugins, install new fonts and do other changes as per table 3. Just for statistical purposes, they collected other tracking information to observe how constant or changeable fingerprints were. However, the results are biased as the interactive nature of the Panopticlick website encourages changing the browser configurations. They found that 37.4% of users who revisited the site more than once have more than one fingerprint over time. Following this finding, they conducted an experiment to see if a connection can be made between an older fingerprint and a newer one. They wrote an algorithm to conduct their experiment. The results were analyzed for users who returned to the site 1-2 hours or more after their first visit and who now had a different fingerprint. Their algorithm made a correct guess in 65% of the cases, an incorrect guess in 0.56% of cases, and no guess in 35% of cases. "99.1% of guesses made were correct, while the false positive rate was 0.86%". The algorithm was not enhanced; hence, with some enhancements the results could be improved.

3.2.5 Convergence of Anonymity, Pseudonymity and Identifiability

We saw how users can be tracked both verifiably (through HTTP cookies and supercookies) and non-verifiably (fingerprinting). The question now arises: can

pseudonymous data be identified. A mid 2011 study done by Arvind Narayanan [40] shows that there are several ways pseudonymous data can be identified. For instance:

1) *The third-party is sometimes the first party*: Some of the companies that are third-party trackers often have a first party relationship with each other. An example of such a site is Google and one of its advertising wings, DoubleClick. When you log into these sites, there is no technical barrier for them to analyze the pseudonymous information they collected from you and map it to your identity.
2) *Leakage of identifiers from first-party sites to third-party sites:* There are various methods where an identity is either unintentionally or intentionally leaked. An example of unintentional leakage would be a user who is logged into a first party site, and has his email in the URL. The referrer header would then contain the user's email. The following example by Narayanan illustrates this fact:

   GET http://ad.doubleclick.net/adj/….
   Referer: http://submit.SPORTS.com/…?email=jdoe@email.com
   Cookie: id=35c192bcfe0000b1…

3) *The third-party buys your identity:* As the name suggests, some first party sells a user's identity to third-party sites. There are advertising data providers that purchase identifying information (e.g. Datalogix [44]) to use in targeted advertising [2].

3.3 Conclusion

In this chapter, we examined how OBA came to be. As we examined, there are many supercookie technologies and even other means of tracking like fingerprinting. Supercookies are generally technologies that can be traced and verified. However, as one method is traced and found, another one emerges. Further exacerbating the issue is the

fact that tracing via fingerprinting is not traceable. McDonald and Cranor concluded in [34], that unless the bigger picture is seen and solved, we risk an arms race with trackers. In the next chapter, we will examine different approaches taken to solve the privacy concerns of users. As we do not intend to kill the online advertising industry, we examine the different solutions with an eye on trying to keep online advertising alive. Furthermore, ProfileMeNot's solution will evade an arms race with evolving tracking technologies, as it simulates a user's browsing behavior. By doing this it will *fool* every existing tracking technology to believe that a user has gone to a particular website. This will happen simultaneously while not entirely killing a source of revenue for publishers.

4 Existing Solutions to OBA and Third-Party Tracking

In order to place our work within other research that is related to this subject matter, we examine different approaches taken to address users' privacy concerns with regards to third-party tracking and OBA. In this chapter, we will investigate the solutions that either exist or have been proposed. We examine each solution's strengths and weaknesses and explain how ProfileMeNot can improve their weaknesses. The solutions that exist mostly try to address the privacy concerns that pertain to online behavioral advertising. However, as we saw in previous chapters, the root of OBA is third-party tracking. As a result, solutions that address OBA, but do not address third-party tracking are not enough to solve the privacy concerns of users.

4.1 Opt-outs and Self-Regulation

In this section we have examined the effectiveness of opt-out cookies and self-regulatory bodies. We will walk through their history and also examine their technical issues.

4.1.1 Opt-outs

As we saw in Chapter 2, privacy in tracking and OBA is a concern for most users. As a result of these concerns, industry groups have provided opt-out options where consumers can state a preference for not receiving behaviorally targeted advertisements [25]. Placing a cookie on your browser that lets them know that you do not want to be tracked usually achieves this. However, the pitfall with this mechanism is that if a user deletes their cookies, they will also delete the opt-out cookie. Secondly, there are many third-party trackers. As we saw in Chapter 3, there were 600 third-party services that placed cookies

on the top 100 sites. Opting out of each advertising company and third-party tracker would be very time consuming. Further complicating the issue is the fact that many of the smaller companies are not known and not all of them offer opt-out cookies. There isn't any universal opt-out mechanism that allows users to permanently opt-out of every third-party tracker. Recently, the Do Not Track project aims to create a "Universal Web Tracking Opt Out" option. We will learn more about the details of this project shortly. As we saw, there is also no law or regulation that enforces a universal opt-out.

4.1.2 Self-Regulation

Self-regulation can be defined as "a regulation system in which business representatives define and enforce standards for their sector with little or no government involvement." [45]. There are some arguments for and against self-regulation. Those in favor of self-regulation argue that this "method will institute protective standards while avoiding the pitfalls of government regulation." [45]. They also argue that self-regulation is less costly while focusing resources on areas where regulations are needed. In the US, the Clinton administration said, "… For electronic commerce to flourish, the private sector must lead. Therefore, the Federal Government should encourage industry self-regulation wherever appropriate… ". However, critics of self-regulation argue that firms will put their own profits ahead of the public interest. As put in [14], "in a capitalist economy such that of the United States, companies tend to resist any restriction at all on actions that promise profit. Leaving it to the private sector to lead the way towards restraint on access to personal data, without at least some prodding, is like leaving it to the proverbial fox to guard the henhouse."

4.1.2.1 Online Privacy Alliance

In 1998, the FTC in the US threatened that if there is no self-regulation forthcoming with respect to online privacy, that it will regulate it itself. This led to development of the Online Privacy Alliance (OPA). The group included lead companies in the industry such as AOL, IBM, HP and others. The goal of creating such an alliance was to create an environment of trust among users and the protection of their information in the online world. The OPA issues guidelines to participating companies. These guidelines include creating a privacy policy that informs users about data collection and use of personal data and also asks for companies to allow users to *opt-out* of those uses. However, the OPA guidelines failed. Only about a hundred companies joined the alliance. The OPA didn't limit the collection of sensitive data or protect users from the harmful use of their data, except through an opt-out mechanism. After a few years, the OPA admitted that it had come up short and began supporting online privacy legislation [45].

4.1.2.2 Network Advertising Initiative and Digital Advertising Alliance

In 1999, after the failure of the OPA, the Network Advertising Initiative (NAI) was created. The NAI was created to exclusively cater to the network advertising industry. Initially the NAI focused on the merging of Personally Identifiable Information (PII) with Non-Personally Identifiably Information (non-PII). The NAI principle required that websites include a privacy policy that will state the practices of a site regarding the collection of PII and how that information would be communicated with third parties. Given that most readers fail to read privacy policies, the NAI requirements for meeting Fair Information Practices requirements were made ineffective. In their principles, the NAI had indicated that through a third-party, it would enforce its principles through

random audits and would sanction non-compliant members. However, by 2003 the organization had failed to accomplish this task and its membership had fallen to two companies. Given their failures the NAI revised their principles in 2008. The initial principle would have third-party enforcement removed and indicated that it will police compliance itself. This latter fact supports the argument that proponents of government regulation suggest, which is self-regulation will not be enforcing enough. The Digital Advertising Alliance (DAA) is also similar to the NAI [45].

Recent trends with the NAI show an improvement in compliance with stopping OBA. Currently, NAI requires its members to give the users either the choice to opt-in or opt-out. This has to been done through an HTTP cookie. The NAI provides users with the opt-out cookie. The opt-out cookie indicates that participating members cannot target behavioral advertising to users who have installed it on their computer. Users can visit the Opt-Out page of the NAI to opt-out of all participating members [46]. In 2010, NAI members confirmed that they are not using Flash cookies for OBA [36]. Rebecca Balebako et al. [25] found that the NAI opt-out mechanism is indeed effective in blocking behaviorally targeted Google text ads. Their study is based on measuring the targeted ads displayed with and without opting-out of NAI and DAA compliant websites. As they mention in their work, the fact the targeted advertisements are not shown does not necessarily mean that a user is not being tracked and profiled.

However, matters are not as rosy with the NAI and DAA compared to recent trends. The study conducted by Rebecca Balebako et al only studies Google text ads and not other companies. Secondly, as acknowledged by Rebecca Balebako et al, users can only opt-out from companies that participate and respect the self-regulation. A third issue with

self-regulation, as noted by Rebecca Balebako et al. with both the NAI and the DAA is that compliant companies only promise to stop delivering targeted advertisements and do not make any pledge with regards to tracking. A study by Jonathan Mayer [61] done on 64 NAI members, found that 32 of them left tracking cookies in place after opting out of behavioral advertising. He also found that seven NAI members that pledged to stop tracking left tracking cookies in place after opting out. On the other hand, they did find 10 NAI members who delete their tracking cookies upon opting-out given that they only promise to stop behavioral targeted advertisements. A fourth problem with self-regulated opt-out is their reliance on cookies. If a user deletes his/her cookies, then they would have to re-install the opt-out cookies they had initially installed. A fifth caveat with self-regulation is that the users have to periodically check for new members joining an alliance or initiative.

ProfileMeNot will not discriminate against any NAI, DAA or non-self-regulatory complaint websites while allowing targeted ads to be displayed to the user. If NAI or DAA members choose not to display targeted advertising, they can still track users and collect their information. ProfileMeNot in either case will introduce noise into their built profile. Even when users delete their opt-out cookies, or there are new NAI or DAA members, ProfileMeNot users do not need to update a list of websites.

4.1.3 Do Not Track

Do Not Track is a project maintained by Stanford Researchers. Do Not Track is a "technology and policy proposal that enables users to opt out of tracking websites they do not visit, including analytics services, advertising networks, and social platforms" [47]. Do Not Track is achieved by signaling the user's preferences with an HTTP header.

Currently Firefox, Internet Explorer and Safari support Do Not Track. However, the number of third parties that comply with Do Not Track is limited [47]. The W3C, World Wide Web Consortium, has released a technical standard draft for the header technology [25]. Their latest working draft can be found at [48].

Do Not Track appears to be a good solution, as it will facilitate opting out of tracking. Furthermore, as Jonathan Mayer and Arvind Narayanan have written to the FTC, Do Not Track would not kill behavioral advertising, but would only cap it. Advertising-supported websites can ask Do Not Track users to allow for third-party tracking to support their business, or even require them to do so. In such cases, the user can choose whether or not they want to use their services or not. Do Not Track can also be extended to mobile platforms [49].

However, as noted, the number of third-party tracking companies that currently honor Do Not Track is only 20. Some major players in the online advertising industry, such as DoubleClick, are missing from the list of those who abide by Do Not Track. The study done by Rebecca Balebako et al., published in 2012, found that Do Not Track headers do not currently block targeted advertising effectively. Do Not Track is also currently not legally binding.

ProfileMeNot does not need to wait for third-party advertising companies to join in a policy to enhance privacy. We do believe that the Do Not Track option could be more effective if it is widely adopted and legally enforced. In this regard, we believe that ProfileMeNot, if widely used, will lead third-party trackers and Web analytics companies to adopt a Do Not Track option in order to get more accurate results.

4.2 P3P: The Platform for Privacy Preferences

Website privacy policies are often very long and not easily understandable. As users are concerned with their privacy and how their information is being used, the World Wide Web Consortium (W3C) developed P3P as a standard way for websites to communicate to users about their privacy policies. P3P's objective is to create a machine-readable privacy policy that can be identified automatically. It also automatically compares privacy policies of a website with a user's predefined preferences and informs the user so they can make appropriate decisions and take the necessary actions. Note that P3P is not an anonymity tool, but rather a tool that increases transparency. P3P tools allow users to set their privacy preferences on their browsers, and have the browser warn them if a P3P-compliant website is not compliant with their privacy preferences [50].

A company's data practices refer to what a company does with data collected from the users. The data practices are often in a very long format, which is very time-consuming and confusing for users to follow. To further complicate the issue, different companies' privacy policies are not coherent or standardized. P3P is a standard vocabulary for describing these data practices and describing the different kinds of information collected. Basically, a P3P policy is a set of answers to a few multiple-choice questions. This type of standard allows P3P to be processed automatically. Using standard HTTP requests, P3P also includes a protocol for requesting and transmitting P3P policies. This helps in identifying P3P-compliant websites. P3P user agents use standard HTTP requests to fetch a P3P policy reference file. The policy reference file is used to specify the location of the P3P policy for each part of the website. Hence, the P3P user agent can fetch the required policy and inform the user.

There has been much effort put into improving P3P [51] [52] [54]. Work is still being carried out in P3P [53]. However, P3P has not had not been widely adapted [54].

P3P policies are not legally binding. Although levis.com, at the time of this case study, was not compliant with P3P, it did have a privacy policy that it did not honor. Levis.com is the eCommerce website of the Levis clothing line. In 2009, a case study was done on the privacy policy of levis.com [56]. On their website, Levis had provided a privacy policy, ``Levi Strauss & Co.`s Commitment to Privacy`` which at the time of the paper [56], had been updated on May 22, 2006. In their privacy policy they pledge not to share any personal information. It also mentioned that they:

"Automatically collect and store certain other information to enable us to analyze and improve our websites and to provide our customers with a fulfilling online experience. For example, we collect your IP address, browser information and reference site domain name every time you visit our site. We also collect information regarding customer traffic patterns and site usage"

A third-party advertising company is also noted in their privacy policy, Avenue A. Avenue A collects "anonymous information about your visits to our website. This is primarily accomplished through the use of a technology device commonly referred to as a Web beacon. Avenue A may use anonymous information about your visits to this and other websites in order to provide ads about goods and services of interest to you". They conclude by mentioning that "By using our website, you're agreeing to let us collect and use your non-personal information as we described in this Privacy Policy". There were no other references to any other third-party data collectors with the exception of Avenue A.

Using Mozilla Firefox's TamperData plugin and examining cookies that resulted in visiting levis.com, Catherine Dwyer found a total of nine Web beacons loaded on the client machine. All nine beacons had P3P policies. She found that these nine beacons linked to eight digital advertising entities. Furthermore, eight of the beacons were used for customer tracking. Further exacerbating the issue is that one of the beacons, tracking.searchmarketing.com, collects identified data such as contact information. As mentioned before, this contradicts Levi's privacy policy that it will not share personal information without consent [56].

The results of the work done suggest that the privacy policies of websites are not always accurate. This goes in-line with users who are against self-regulation and leaving it to the industry to comply. ProfileMeNot takes a different approach to privacy. ProfileMeNot does not require putting its trust in any privacy policy. However, using P3P and ProfileMeNot together are not mutually exclusive. P3P implementations can be used in conjunction with ProfileMeNot.

4.3 Blocking

Another avenue of solution is blocking technologies. Blocking technologies prevent tracking and online advertising by either blocking all third-party cookies and scripts or only those from a specific domain or list. These tools can be built into the browser, such as Firefox’s third-party cookie blocking. We generally oppose solutions that block advertisements. However, in this section, we will show that blocking advertisements does not necessarily block tracking.

4.3.1 Complete Blocking

The goal of our work is not to totally kill online advertising as providing advertising space to advertisers funds many websites. The issue with tools such as AdBlock Plus and similar technologies is the fact that it blocks all ads, hence killing a source of revenue for websites that depend on funding from advertisers to operate. Also, as we saw in Chapter 3, technologies such as AdBlock Plus try to block scripts, Flash, iFrames etc. As we saw in the development of tracking technologies, trackers will find innovative ways to circumvent the limitations set by technologies such as AdBlock Plus. AdBlock plus does acknowledge the fact that many websites operate on revenue generated from ads, and hence created a whitelist of accepted ads with a few strict criteria. Although this might seem like a useful initiative, it still will block many ads that do not meet their criteria. Note that AdBlock Plus will indiscriminately block all advertising, whether contextual, location-based or targeted ads based on a user's inferred interest. Another issue with blocking tools such as NoScript, is that they considerably reduce the user's experience. NoScript allows "executable Web content such as JavaScript, Java, Flash, Silverlight, and other plugins only if the site hosting it is considered trusted by its user and has been previously added to a whitelist" [57].

ProfileMeNot allows for all different types of advertising to be displayed, while at the same time not needing to catch up with different tracking technologies. It also does not require a whitelist of allowed advertisements, as it will allow every type of online advertisement to be displayed and still maintain a user's privacy, hence rendering it more usable.

4.3.2 Blacklisted Blocking and Partial Cookie Blocking

Another category of blocking tools prevent third-party advertising and tracking by refusing content on a blacklist from specific domains. An example of such a tool is TACO 4.40 developed by Abine. TACO 4.40, as claimed on their website, blocks blacklisted third-party HTTP cookies, Flash cookies, DOM cookies and a few more. However, they cannot block browser fingerprinting, as fingerprints don't leave any trace on the client computer. In the study done by Balebako et al; they found the blocking tools they tested to be effective. However, as technologies such as TACO block tracking, trackers, as we saw in Chapter 3, will find innovative ways to circumvent any third-party blocking tool. Furthermore, they report that only around 5-18% of Internet users block third-party cookies.

A study conducted by Leon et al. [58] tested nine tools. The tools either blocked access to advertising websites or set cookies indicating a user's preference to opt out of OBA; some were tools that were directly built into the browser. They found usability flaws in all nine tools. The users participating struggled installing and configuring blocking lists. Another major issue with most of the tools is their default settings. For example, Ghostery and TACO do not block any trackers by default; however, the user installing them is indicating that they are not interested in being tracked.

ProfileMeNot, as we explained in the previous subsection does not need to be in an arms race with trackers and their tracking technologies, whether they are traceable or not. Furthermore, ProfileMeNot is easily installable, and once installed, does not need any configuration by default.

4.3.3 Private Browsing Mode

The goal of private browsing is to protect the data of users and browsing habits from two types of attackers; the *local attacker* and the *Web attacker*. The local attacker is an attacker who gets access to a machine that a user was using for private browsing at time *T*. The goal of private browsing is that the attacker should not learn any information about private browsing actions prior to *T*. The second goal of private browsing is to not allow Web attackers to link private browsing and public browsing sessions together. A study done by Guarav Aggarwal et al. [55] found that browsing in private mode is not consistent among the four major browsers: Firefox; Chrome; Safari; and Internet Explorer. Furthermore, they found some security breaches that can allow determined Web attackers and local attackers to obtain the information they need. In Firefox, they located 24 points in Firefox 3.6 code that control all rights to sensitive files. With respect to a Web attacker, they found that some of the points in the code didn't adequately check for private browsing sessions; however, most did. Of the ones that didn't check were security certificate settings, which also included SSL client certificates. The certificates can be either manually imported by the user or automatically done by authorized websites. They also found that Firefox extensions with binary components are all unsafe since they can read or write to any file on the disk. Of the 32 JavaScript-only extensions that they examined, they didn't find any violations in 16 of them.

The experiments done in [55] were mostly on Firefox, since the code is open source. Google Chrome's Incognito mode does not many make any claims that it will block third-party trackers from tracking users across the Web. Choosing the incognito mode in

Chrome opens a new window and displays Figure 3 which clearly specifies that the incognito window doesn't stop tracking.

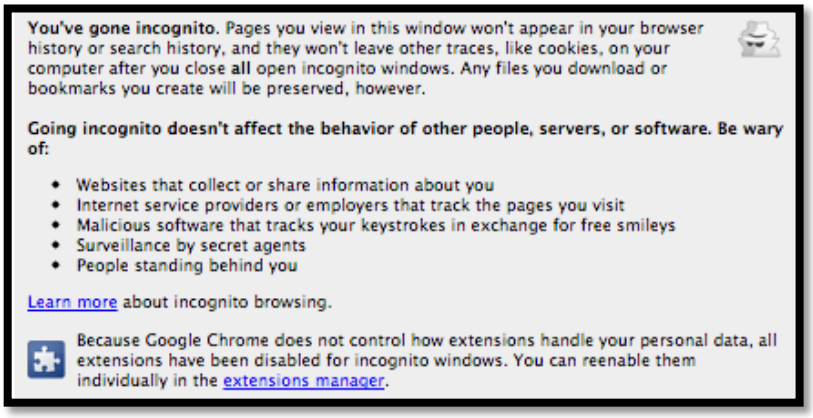


Figure 3: Google Chrome`s Incognito mode message upon opening a new Incognito window

ProfileMeNot does not discriminate against private browsing mode and public mode as it causes noise and obfuscates the tracker's profile in whichever method they use to track users.

4.4 Proxy-based Anonymizers

Generally, anonymizers access the Internet on your behalf. Anonymizers are usually based on a proxy-solution. Proxies require users to grant some trust to a third-party. The third-party could either be a server or a final node in a TOR (The Onion Router) type of system. Proxy-based solutions have previously been shown to be vulnerable. In 2007 a Swedish Security consultant intercepted usernames/passwords of email accounts by operating and monitoring TOR exit nodes. TOR cannot encrypt traffic files between an exit node and the user requesting a particular page. In such cases, trust must always be put in the exit node or any other intermediary node where the data is not encrypted. TOR-like system users could also potentially be blocked from accessing some sites.

Further exacerbating the use of proxy-based solutions is the fact that they are generally not easily configurable [12].

ProfileMeNot users, while using a static IP address, might also be blocked. However, as ProfileMeNot visits many websites randomly, and the algorithms try to simulate an actual user's behavior, the probability of it happening is low. If a website blocks requests from a particular TOR exit node, it will affect many users, whereas ProfileMeNot will only affect one user while using a given static IP address. Furthermore, ProfileMeNot does not require any trusted third-party, such as a proxy, exit or any other intermediary node. In contrast to proxy-based anonymizers, ProfileMeNot is easy to install and use without further technical knowledge.

4.5 Solutions that allow both Privacy and OBA

Several architectures have been proposed to allow both OBA and privacy. These solutions require changes or additions to current protocols. We will examine Adnostic [1]. Both RePriv [60] and Adnostic profile the user on the client-side. In the case of Adnostic, ad-selection is also done on the client-side and in RePriv; information is selectively shared with advertisers and recommendation systems upon a user's consent.

4.5.1 Adnostic

In a nutshell, Adnostic is an architecture that allows for targeted advertising without compromising users' privacy. This is achieved by profiling the user on the client's-side rather than the trackers. Ad-networks/Ad-servers send n ads to the browser, and the browser, based on the client's behavior, selects the appropriate ad to display. The choice of n can vary. A challenge in this architecture is billing. The charge-per-click model of advertising stays the same, but the charge-per-impression model changes since this

information is to be private. Ad-networks need to know which advertiser to bill for the displayed ads without knowing which ad was displayed to whom. This problem is solved by using a additive homomorphic encryption scheme and zero-knowledge proof. The incentives of deploying Adnostic for publishers and trackers are the following: targeting in private browsing mode; standardize audience segmentation; and potentially improve user tracking. Given that users can synchronize their browser preferences on different computers, Adnostic can help enhance targeted advertising by consistently profiling the user across different computers, whereas current ad-networks are unable to do so. Adnostic keeps ClickStream, Behavioral profile and ad impression history private. It does not keep ad click history private and argues two reasons for this: defense against click-fraud, and the fact that advertisers see clicks on their ads and have a strong incentive to share this information with ad-networks. One issue that might stop some advertising networks from adopting Adnostic is their proprietary tracking and targeting algorithms. Ad-networks may not be satisfied with the ad selection process of Adnostic.

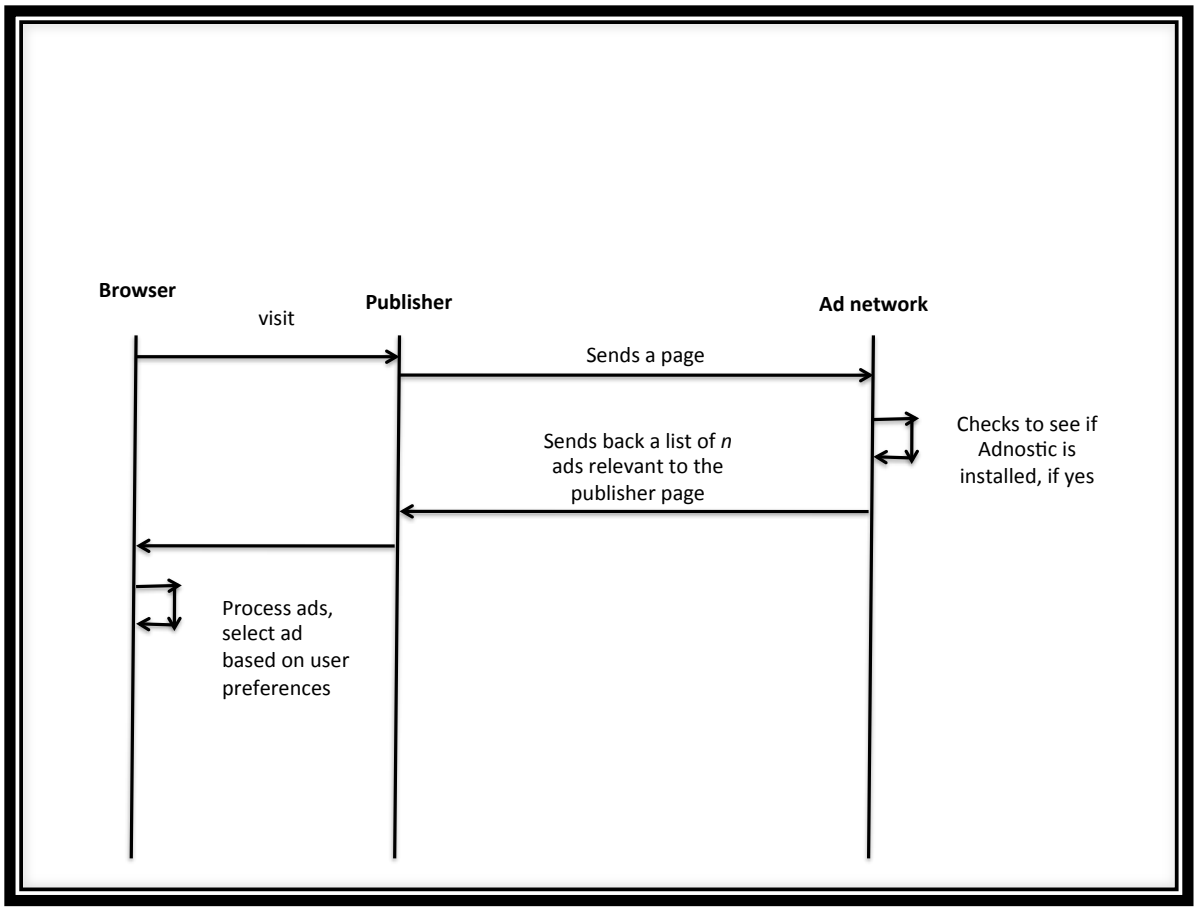


Figure 4: Adnostic architecture

Figure 4 gives a graphical representation of how Adnostic functions.

Adnostic is developed as a browser extension. Currently, it is developed as a Firefox extension. We believe that Adnostic is a suitable replacement for tracking. It allows for both privacy and targeted tracking. However, ad networks and publishers, to the best of our knowledge, have not been adapted. ProfileMeNot, like Adnostic, is a client-side solution that allows for OBA. Unlike Adnostic, ProfileMeNot does not require any publisher, ad-network, or tracker to change or add a new feature to its existing protocols. Given the noise introduced in ProfileMeNot, it will affect the data that third-party trackers gather. A widespread use of ProfileMeNot could lead the trackers to being more

privacy-conscious, perhaps by adopting an Adnostic-like approach. Furthermore, Adnostic does not solve the issue of third-party tracking for other trackers, such as analytics companies. We believe that further enhancement is needed to the protocol to completely remove the need for third-party tracking.

4.6 Conclusion

In this chapter we examined existing technological solutions and policies proposed to solve the privacy concerns of users. We concluded that no existing solution is either comprehensive enough or widely adopted by advertising networks. In the next chapter, we will introduce another class of solution: achieving privacy through obfuscation.

5 Obfuscation, TrackMeNot and ProfileMeNot

In this chapter, we will first introduce the concept of obfuscation as a means of defense. We then introduce and examine TrackMeNot in depth. After introducing TrackMeNot, we introduce our work, ProfileMeNot. ProfileMeNot is a solution that uses obfuscation as a technique to achieve privacy against third-party trackers. ProfileMeNot is implemented as a Firefox extension that is built on top of TrackMeNot [12]. TrackMeNot is designed to achieve privacy by obfuscating a user's Web search, in order to create incorrect and misleading information to third-party trackers in the interest profile that they build for users.

In short, we are taking the concept of obfuscation [14] and have built ProfileMeNot based on a similar idea for web-search obfuscation [12]. Our work obfuscates a user's Web browsing history on top of their Web search history. In this respect, our work differs from TrackMeNot. We also evaluate our results through a group of volunteers and try to quantify our findings. TrackMeNot very rarely visits websites; its intent is to obfuscate the Web search queries of a user and not their Web browsing history. To the best of our knowledge, no one has yet created a tool or architecture to use obfuscation to achieve privacy in Web browsing habits and history. Furthermore, TrackMeNot does not quantify the amount of noise it introduces. ProfileMeNot does not aim to abolish online advertising. ProfileMeNot will have no impact on advertisers who pay per click; however, it will have an impact on advertisers who pay per impression. We hope that our solution encourages the online advertising industry to adopt a guaranteed privacy-preserving model. Throughout our evaluations, we aim to reach an upper bound of 100% noise on the behavioral profile of a user. By 100%, we mean to introduce the same

amount of fake interests as a user has real interests. This percent of noise will then keep a balance between the users and profilers. Profilers can then know with a probability of ½ if a user's interest is real or fake. Additionally, introducing too much noise will increase the likelihood of profilers and trackers identifying the presence of an automated system, and so 100% seemed like a reasonable choice.

5.1 What is Obfuscation?

Obfuscation, for our purposes, can be defined as "producing misleading, false, or ambiguous data to make data gathering less reliable and therefore less valuable" [14]. Obfuscation is a method that has been deployed in many other domains**.** Brunton and Nissenbaum investigate obfuscation in depth within different aspects of computer-enabled data collection [14]. In World War 2, both the British and the Germans developed a system that would feed the adversary's radar systems false information regarding the number and position of fighter crafts attacking a city, called the chaff. The German Luftwaffe and the British RAF (Royal Air Force) both developed it independently and both were worried that if they start using it, the other side would quickly learn the trick as well. The systems were identical: small aluminum strips that created false echoes for the radar were dropped every minute from the airplanes. Hence the adversary couldn't distinguish between false plane locations and real locations. This helped the British in bombing Hamburg in 1943. This is an example of obfuscation, where creating false data makes the correct data useless or hard to distinguish. As a matter of fact, most military aircrafts and warships have a chaff dispensing system for self-defense. The basic idea of all obfuscation is pretty much the same: create random false data to make it indistinguishable from correct data [59].

A more relevant example of obfuscation is FaceCloak [3]. FaceCloak protects the privacy of the users on a social networking site by shielding a user's personal information from the site and from other users that were not explicitly authorized. FaceCloak stores sensitive information on a separate server and provides fake information to the social networking site. FaceCloak lets users pick which information they would like to keep private and which they would like to keep public. The information that the user wishes to keep private from Facebook (e.g. age, gender, political views) is encrypted and stored on separate servers and are only decrypted by authorized friends who have the key to decrypt the information. For the information to be kept private, it generates fake information (such as random age etc.). In this manner, information is kept private from both Facebook and unauthorized users. This type of obfuscation is called *selective obfuscation* [14].

5.2 TrackMeNot

TrackMeNot (TMN) is a lightweight Firefox extension that tries to achieve privacy in Web searches "by obfuscating users' queries within a stream of programmatically generated decoys" [12]. TMN does not require users to rely on any third-party and runs solely on the client's side. TMN does not randomly select queries. "Query-like phrases are harvested by TMN from the Web and sent, via HTTP requests, to search engines specified by the user." [12]. The queries are sent by TMN on an average frequency set by the user. It also tries to mimic an actual user's behavior. The following subsections are the measures TMN has taken to achieve its goal of hiding users' Web searches and also simulating an actual user's behavior. As ProfileMeNot is built on top of TMN; we will

mention which sections of TMN have remained untouched and which have been changed.

5.2.1 Dynamic Query List

Upon installation, TMN creates an initial seed list of query terms from RSS feeds of different websites. The user can set the RSS feeds. The default RSS includes feeds from nytimes.com and theregister.co.uk. Queries are also taken from a list of popular queries gathered from publicly available lists of recent search terms and are added to the initial seed list.

As TMN continues to run and performs searches with the queries built in the list, it randomly picks individual queries and marks them for substitution. Once a marked query is scheduled for search, TMN intercepts the search engine's HTTP response and through a series of regular-expression tests, attempts to find a query-like term from the HTML returned. If it finds a term successfully, it then substitutes the new term with the marked term for substitution in the list and the substitution mark is removed. Every time the browser is started, an RSS feed is randomly selected from the list of RSS feeds that are either there by default or have been added by a user. A subset of its terms is substituted into the seed list in the same randomized substitution method previously described. Figures 5 and 6 show a seed list of queries that have changed over several weeks. The figures are constructed based on an example in [12].

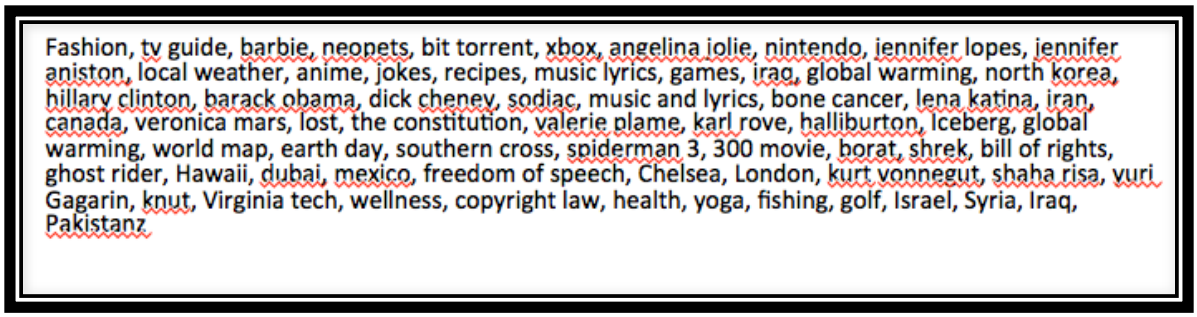


Figure 5: Query seed list [12]

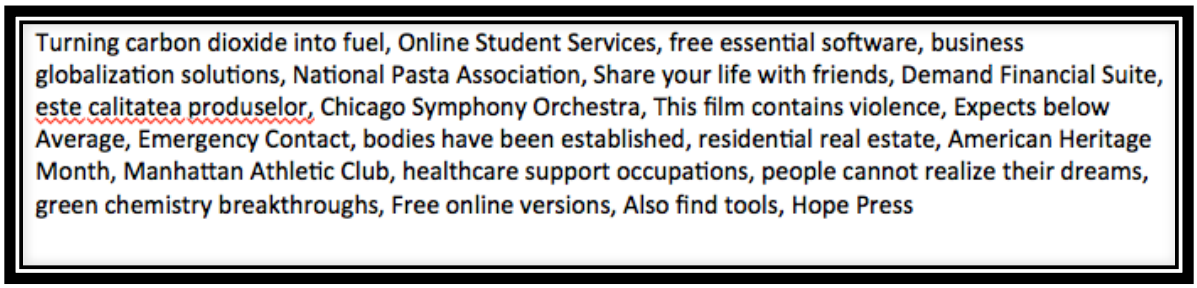


Figure 6: Query seed list from Figure 5 having evolved after a few weeks [12]

Partly on the random selection of queries for substitution, partly on new terms gathered from RSS feeds every time the browser is started, partly from nondeterministic query extraction from HTML responses, and in part from the changing nature of Web search results, each user using TMN will develop a unique list of query terms. The dynamic list of queries helps in evading filters [12]. ProfileMeNot keeps this aspect of TrackMeNot but with some added default RSS feeds. A contributing factor to choosing TMN to build our work on top of was the dynamic query list. With the dynamic query list, new links appear. New links help simulate an actual user's behavior more closely, as a user does not only visit a static list of websites.

5.2.2 Selective Click-through

TMN currently employs *selective click-through*. After a series of regular expression tests are done, the resulting web pages from a search query and one or more link is chosen to simulate clicks on. The regular expression tests are there to avoid clicking on revenue generating ads. The developers of TMN have also tested "Click-Through," in which TMN tries to follow one or more additional links on a results page after an initial query. Given that the intention of ProfileMeNot is to obfuscate a user's Web browsing history, we have removed this part of TMN. TMN employs selective click-through with a very low probability (based on empirical analysis). ProfileMeNot uses a different approach. One major contributing factor to implement our idea of obfuscation on top of TrackMeNot was TMN's existing selective click through. We reverse engineered their code and found where the search engine results were being parsed.

5.2.3 Real-Time Search Awareness

In order to mimic a user's functionality better, TMN finds out when a user has initiated a search on one of the engines that TMN is using. This aspect of TMN has remained unchanged. However, since we only managed to reverse engineer how TMN parses the Bing search engine's search results, ProfileMeNot's functionality currently only works with Bing. Given that Bing's market share is 16.5% [41], it is unlikely that this functionality is utilized in ProfileMeNot.

5.2.4 Live Header Maps

To mimic a user more accurately for each search engine, TMN stores a set of variables that represent the header fields that were used when the user last carried out a search on a search engine. In this manner, TMN can produce the same set of header maps when it sends search requests. Moreover, TMN uses the last URL to access a search engine. For example, if a user used a search engine's toolbar to send a search query or if they used the website of the search engine, the URL would be different. TMN can further change the header maps dynamically as the user performs search. ProfileMeNot did not try to change this.

5.2.5 Burst-Mode Queries

TMN gives users the option to set the frequency of the number of searches to be performed. Whichever interval is chosen, the user has two options. The first option is to perform each search with semi-random intervals, with the average being set by the user. By semi-random, it is meant that the queries are sent in random intervals from each other, but the average number of queries sent within a time frame (minute or hour) is fixed. The other form is to perform a batch of queries close to the actual user's search. The Real-Time Search Awareness feature mentioned earlier detects the search timing. In this mode, a subset of queries is permuted to create a "search theme". For example, the query "dancing with the stars" was permuted with "stars", "dancing with", "with the stars", and "dancing with the stars". The goal of the burst-mode is to further mimic an actual user's behavior. ProfileMeNot keeps the option enabled by default for two reasons. The first reason is the same as the reason in TrackMeNot: to mimic a user more. As some words

are repeated in a sequence of searches, the second reason we enabled burst-mode is to have ProfileMeNot visit more than one page in the same interest category.

5.2.6 TMN Control Panel

Through the control panel, the user can set the frequency of the Web searches. They can also pick which search engines they would like TMN to use. The TMN frequencies are in units of *queries per minute* or *queries per hour*. TMN also allows users to enter a list of blacklisted keywords. These keywords will then not be queried. ProfileMeNot keeps most options of the TMN control panel, with some minor changes. ProfileMeNot, for the purposes of our testing, has only two options available. The first is *one link per minute* and the other is *30 links per hour*. We use links per minute/hour instead of queries, as we our main focus will be on Web history rather than search query history. We also limit the users to only Bing in our experimental studies.

5.2.7 Implementation Details

TrackMeNot is developed as a Firefox extension. We took around two weeks to learn how to make Firefox add-ons, and an additional three months to reverse engineer TrackMeNot. We did not reverse engineer the entire code, only enough to reach our goals in ProfileMeNot. We present the implementation details of TrackMeNot 0.7.82. We reverse engineered and integrated parts of TrackMeNot 0.6.728. There were some major changes between the two versions. We did not manage to find out how the returned Bing results were parsed on the later version; hence we integrated where that happened in 0.6.728 into TrackMeNot 0.7.728.

After creating the initial seed file based on debugging TrackMeNot, we found that TrackMeNot, once enabled, schedules a search right away. After that, it has a timeout.

The timeout is dependent on the value set by the user and some other parameter. Even though the user can select 10 queries per minute as the average frequency, the frequency rate does not go over three frequencies per minute. Javascript has two timing event functions: setTimeOut() and setInterval(). The difference between the two is that setTimeOut schedules an event only once after a period of time, whereas setInterval() schedules an event periodically. We noticed that TMN never calls a setInterval(), rather it calls setTimeOut(). This would mean that there is some sort of a cycle of calls that is happening within the code. After further reverse engineering, we found there are two events that when they occur, trigger a call to an observe method. These two methods were "http-on-modify-request" and "http-on-modify-response". On most occasions, if there is not an error, TMN schedules another search when the HTTP response from the previous search is okay. Before the next search is scheduled, there is the aforementioned timeout. The timeout tries to reach an average rate by either performing searches closer to, or more distant from, each other. For example, if the rate is set to one query per minute, it might submit two queries 30 seconds apart, and the next two 90 seconds apart. If there is an error the timeout will be set for a longer duration. There are some other options that can be scheduled.

TMN has 6 main JavaScript files. The file *trackmenot.js* is where TMN creates its main frame and also controls most of the flow of the program. Other files are helper files. The main frame is the frame where TMN operates its searches. It hides the frame by default, but the user can make it visible and see what TMN is doing. The user can also see the queries that TMN is sending via a status bar on the bottom left corner of their browser.

The main file that deals with search functions is *tmn_search.js*. It has the function *doSearch()* which requests a query to be sent to a search engine.

The file 'rep_proceessor.js' is the file that has functions to parse and process returned links in 0.6.728. We took the *rep_processoer.js* of 0.6.728 and integrated it into version 0.7.728. The file in 0.6.728 includes functions that extract the links that have been returned from the search engine. It also filters some words such as "Images, News etc." that are in the title of the link. This is done in order to filter default links that are returned by the search engine. For example, the link identified with the word *Images* would take the user to www.bing.com/images.

The main challenge in understanding their code was finding out how the code functioned and which calls were made where. We had to initially learn to build Firefox extensions and familiarize ourselves with Firefox's functions. We also utilized information in [62] to help us reverse engineer the code.

5.3 ProfileMeNot

ProfileMeNot aims to achieve privacy in a Web user's browsing history from third-party trackers by obfuscating the user's search history. Our particular goal is to create noise in the profile that third-party trackers create for users, such that the interest segments that the profilers place users under is not accurate. Our goal is not to totally kill behavioral advertising. Rather our aim is to allow for 100% noise in the behavioral profile of the user, at least at the *parent* layer. We will see what is meant by the parent layer in Chapter 6. In this manner, each element on the profile would be the user's actual interests with a probability of 50%.

5.3.1 ProfileMeNot, Overview

ProfileMeNot achieves Web browsing history obfuscation by visiting pages returned from search engine queries that are returned as a result of how TrackMeNot operates its search query obfuscation. Once the results of a search query are returned, the results are parsed, and based on Algorithm 1 or Algorithm 2, a web page is either visited or revisited. While the queries searched are done in a hidden frame, the same as in the current version of TrackMeNot, our goal was to create a proof of concept. Hence we did not focus much on its usability.

5.3.2 Dynamic Query List

The search queries follow the same mechanism as the dynamic search queries of TrackMeNot. For further details on how TrackMeNot selects its search queries, see section 5.2.1. TrackMeNot, by default, has four RSS feeds that it uses for query extraction. The default four are: nytimes.com, cnn.com, msnbc.msn.com and theregister.co.uk. Given that they are all news sites, we added five additional RSS feeds: cbc.ca/sports RSS feeds, geek.com, style.com, dsc.discovery.com and http://blog.travelchannel.com/man-vs-food-nation/feed/ main RSS feed. With the RSS feeds, we tried to diversify the topics of search.

5.3.3 Burst-Mode

As explained in 5.2.5, TrackMeNot has a *burst-mode* feature. In order to keep the interests of users closer together in a batch of queries, we set burst-mode enabled by default and asked volunteers not to change it for our experiments.

5.3.4 ProfileMeNot General Filters

As ProfileMeNot visitation algorithms take results from TrackMeNot's querying the search engines, not every site is an adequate site to visit. While extracting results from the HTML returned by the search engine as a result of a search query, TrackMeNot has readily built filters to filter non-search-related terms. For example, the Bing search engine, when the HTML returned from a search query is returned, returns www.bing.com/news etc. TrackMeNot already identified these keywords given that Bing.com doesn't change its format. TrackMeNot also filters ads from the extracted results.

Provided that TrackMeNot did not have to visit websites very often, ProfileMeNot takes results from the links extracted by TrackMeNot after its filters and adds additional filters. Throughout creating ProfileMeNot filters, we had four goals in mind: bandwidth consumption, user concerns, and usability issues. In order to address bandwidth concerns, prior to sending out ProfileMeNot for experimentation, we let ProfileMeNot run for nine days. During this time, we observed websites that automatically opened videos and audio files. We tried to find common words that were included in either the title or the URL of the links. For example, we noticed that many webpages that include a video on them have one of the following keywords in their URL: "video, tube, watch, live, clip etc.". We identified these keywords and filtered them.

Prior to starting our experiments with volunteers, we asked the early volunteers to fill out a survey. After having verbally explained to them what ProfileMeNot will do, we asked users about the primary concerns they might have in using such a tool. The predominant concern among users was the type of websites ProfileMeNot would use. The

major concern among those volunteers who filled this survey was visits to sexually explicit websites. We tried our best to filter out words in URLs and links that could be related to such content. We did tell users that we couldn't guarantee this. Fortunately, TrackMeNot also allows users to enter a list of *black-listed* keywords. TrackMeNot would then not query these keywords. As we had identified user concerns, we, by default, added some keywords in areas that users had expressed concern in. ProfileMeNot also filters Wikipedia as we noted that Wikipedia does not have any third-party content on it.

5.3.5 Page Visit Algorithms

In this subsection, we introduce the two algorithms that we experimented with. The experimental procedure and results are in Chapter 6. Algorithm 1 is a basic algorithm. The goal of Algorithm 2 is to simulate an actual user's behavior.

The first algorithm is a very simple algorithm that visits each page it selects only twice and does not intentionally revisit them later. The second algorithm utilizes statistics on the number of page views per minute, revisitation patterns and browsing categories of users. The data is based on results from Microsoft, Yahoo, Google and University of Washington researchers [42] [43]. Although we were certain that Algorithm 1 would produce more than 100% noise in most cases, we wanted to achieve a proof of concept that obfuscation works and also to prove that it is possible to control the amount of noise introduced. The aim of Algorithm 2 is to achieve 100% noise in a user's interest profile, such that an adversary can guess if a user's interests are correct with a probability of $1/2$. Algorithm 2 tries to achieve this while visiting the same number of pages as Algorithm 1 does. An adversary in this context is any third-party tracker. Our hypothesis is that

Algorithm 2 will create a median noise of 100% in our experiments. In this section, we explain in detail what ProfileMeNot does and which features of TrackMeNot it utilizes.

5.3.5.1 Algorithm 1, ProfileMeNot Basic

As described earlier, we developed two different algorithms. In a nutshell, Algorithm 1 extracts the results of a TrackMeNot search, passes them through ProfileMeNot's filters, and then randomly selects a link from the remaining results and visits that link. The next time a search is done, it still revisits older links it visited with a probability of 50%. This is in order to have the tracking companies see that the user has visited a page more than once, hence increasing the probability that they register it as an interest. Once a page has been visited twice, a new page is visited. The frequency of new links visited for experimental purposes was set to either *30 new links per hour* or *one new link per minute*. We will see the rationale behind the frequencies in Chapter 6. In either case, the number of search queries done based on TrackMeNot is twice the amount of new link visits.

Algorithm 1 ProfileMeNot Basic

```
1:  when query q is searched by Bing
2:   filter and extract links returned using TrackMeNot and ProfileMeNot filters
3:    if (there are one or more links left after the filter)
4:       if (first time a link is being visited) or (last link has been visited twice)
5:          Randomly pick a link l from the extracted and filtered links
6:          Visit l
8:       else
9:        Visit link that was visited in the last query searched by Bing
10:      end if
11:   else
12:     return
13:   end if
14: end when
```

5.3.5.2 Algorithm 2, ProfileMeNot Ultimate

Algorithm 2 is a smarter algorithm than Algorithm 1. Algorithm 2 primarily aims to mimic an actual user's behavior. By simulating an actual user, its secondary goal is to create 100% noise in the interest profile that third-party trackers build on users through the means of obfuscating a user's Web browsing history by visiting webpages on behalf of the user. In this way, the data of users is accurate with a probability of 50%. In this context, simulating an actual user's behavior consists of three things: how often users revisit pages, the types of websites that users revisit and how often do users browse the Web. To achieve this, it would have to visit around the same number of pages as an actual user does. Since Algorithm 2 is very complex, we explain it in different levels of abstraction and build up on it. We first present Algorithm 2.1, which is a top-level view of Algorithm 2. We then investigate user Web page revisitation habits and introduce Algorithm 2.2. Algorithm 2.2 is a more detailed view of Algorithm 2.1. We then discover user-browsing habits and map them to Web pages that users revisit. Finally, we present Algorithm 2. We also mention the average time users spend online and explain it in more detail in Chapter 6.

5.3.5.2.1 Algorithm 2.1

In this sub-section we present Algorithm 2.1. Algorithm 2.1 is a top-level view of Algorithm 2. Algorithm 2 is called whenever a query q is submitted through TrackMeNot algorithms to the Bing search engine.

Algorithm 2.1 ProfileMeNot, Pick site

1: **when** query q is searched by Bing
2: filter and extract links returned using TrackMeNot and ProfileMeNot filters
3: with probability P visit a link from the extracted and filtered returned links from Bing
4: with probability $1 - P$, *select* a link from the extracted and filtered returned links from Bing and visit a
previously *selected* link
5: **end when**

As is evident in the algorithm, step 4 does not revisit the links in step 2, but rather selects new ones and revisits them. In our experiments, we set $P = {}^{1}/_{2}$. The choice of ${}^{1}/_{2}$ is based on studies that show page revisitations constitute 50% of users' browsing behavior [42]. This choice is the first step of simulating the browsing behavior of an actual user. Before moving on to what *selected* links are in the Algorithm 2.1, we will give more details about user revisitation patters in order to more closely simulate an actual user's behavior.

5.3.5.2.2 User Page Revisitation Habits

To further mimic an actual user, we conducted more research on revisitation patterns, change in content of pages that are revisited, and the categories of pages that a user visits. In particular, we looked at [42], which focuses on the relation between revisitation patterns and change of page content over time. The authors also survey the intent of page revisitations. The study was based on revisitation patterns of over 40,000 Web pages over

five weeks on analyzing the Web log trace of 2.3 million opted-in users of the Live search toolbar. The reason we chose this work as the basis of our algorithm is the scale of data they had in hand. They observed the pages rather than tracking users. Some of their other key findings in [42] with regards to revisitation patterns were as follows.

- The more popular a page, the more rapidly it changes.
- The more times a page is revisited, the more rapidly it changes.
- Quick revisits are strongly related to change.

In [64], the same authors conducted a survey where they asked users about the intent of revisitations. The survey was conducted on 20 volunteers who were all Microsoft employees. The volunteers were asked to install logging software that logged their Web page visits from 55,000 URLs in their log study and from volunteers cache and Web history. At the end of the log period (around one to two months), each participant was asked to complete a survey regarding ten of the Web pages they had revisited. For each Web page, they were asked if they remembered visiting and revisiting the page. "If they remembered the page, they were asked to indicate their intent from a list of options" [64]. The results of the survey were published in [42]. Of the total Web pages (~160 URLs), 39 of them were remembered. Participants responded that for 19 of the URLs, they were interested in finding new information when they visited the page. For 11 of them, they responded that they were monitoring change, and for 9, they were interested in previously viewed information. Since we did not have any other ratio of the type of sites that users revisit, in our Algorithm, we considered the ratio of 19/39 pages to be pages that users revisit to obtain new information, 11/39 to monitor information and 9/39 are pages where users would like view previously viewed information. They also found that the most

common reason people visit a page is to use a search engine or enter data into a form. Their analysis of revisitation and change goes far deeper. They note from previous research that 50% of all page views are revisitations. They categorize page revisitation patterns into four categories: fast page revisitations, medium page revisitations, slow page revisitations, and hybrid page revisitations. Fast page revisitations are pages that are visited many times over a short period of time (e.g. over the span of a few minutes). Medium page revisitations are pages visited in intervals between an hour and a day. Slow page revisitations are pages that users visit in an interval of a week or more. Hybrid page revisitations are pages that are visited with a combination of fast and slow. While [42] looks for intent and the relation between revisitation and change, it does not focus on the type of pages. Based on the findings in [42], we incorporated the following in the design of Algorithm 2:

1) 50% of page views are revisitations.
2) The more times a page is revisited, the more rapidly it changes.
3) Around 48% (19/39) of page revisits are done with the intent to view new information that is constantly/rapidly changing. Let r represent the percent of pages visited in this category and let $rapidly\_changing$ represent a list that contains links to pages that change rapidly.
4) Around 28% (11/39) of page revisits are done with the intent of monitoring information. Let m represent the percent of pages visited in this category and let $monitoring\_information$ represent a list that contains links to pages that are in this category.

5) Around 23% (9/39) of page revisits are done with the intent to view previously viewed information. Let e represent the percent of pages in this category and let $everything\_else$ represent a list that contains links of pages that are in this category.

Bullet points 3, 4 and 5 are not from the 2.3 million user log statistics, but a survey from a group of volunteers at Microsoft. Although the information we take from them is not their findings in its entirety, we believe them to be sufficient as a first step in simulating an actual user's behavior. The *selected* links in Algorithm 2.1 are links that fall in one of the three lists: $rapidly\_changing$, $monitoring\_information$ and $everything\_else$. We will describe how we selected them after we present Algorithm 2.2.

5.3.5.2.3 Algorithm 2.2

We are now ready to present Algorithm 2 with more details than Algorithm 2.1. Based on the five points in sub-section 5.3.5.2.1, we enhance Algorithm 2.1 and present Algorithm 2.2. Prior to presenting the algorithm, we introduce the variables and notation we used in Algorithm 2 and Algorithm 2.2 in table 4.

Name of Variable	Value	Description
s_1	26	Maximum size of $rapidly\_chanding$
s_2	14	Maximum size of $monitoring\_information$
s_3	13	Maximum size of $everything\_else$
l_1	10	Number of links that can be added to $rapidly\_chanding$ without any extra conditions. After this, links get added to $rapidly\_chaning$ with a probability of p_1
l_2	6	Number of links that can be added to $monitoring\_change$ without any extra conditions. After this, links get added to $monitoring\_change$ with a probability of p_2
p_1	0.5	Probability of a new link that falls under $rapidly\_chaning$ is added to the list once the size of $rapidly\_changing$ exceeds l_1.
p_2	0.5	Probability of a new link that falls under $monitoring\_change$ is added to the list once the size of $monitoring\_change$ exceeds l_2.
rc	0.45	This value is equal to r^*. Probability to visit a page that is in $rapidly\_changing$.
mi	0.75	$mi = m + r$.
$mi - rc$	0.30	This value is equal to m^*. Probability to visit a page that is in $monitoring\_change$. To select which list to visit a page from, a random number is generated between 0 and 1, thus requiring checking for bounds (between $r = rc$ and $r + m = mi$) for the probability to visit a link in $monitoring\_change$.
$1 - el$	0.25	This value is equal to e. Probability to visit a page that is in $everything\_else$. To select which list to visit a page from, a random number is generated between 0 and 1, thus requiring checking for bounds for the probability to visit a particular list.
$P_{revisit}$	0.5	The probability to visit a link and discard it, or to check if a link is suitable to be added to a list (given that there is space) and selects to visit a link in one of the lists.

Table 4: List of variables and notation used in Algorithms 2 and Algorithm 2.2 along with the values we experimented with and selected based on research data
*: For simplicity, we set the values to round numbers

Using the notations in table 4, we present Algorithm 2.2 that is more detailed than Algorithm 2.1. Algorithm 2 is a more detailed version of Algorithm 2.2.

Algorithm 2.2 ProfileMeNot, Pick site

1: **when** query q is searched by Bing
2: filter and extract links returned using TrackMeNot and ProfileMeNot filters
3: randomly pick a link l from the extracted and filtered returned links from Bing
4: with probability $P = 1/2$ visit l
5: with probability $1 - P = 1/2$
6: randomize $p, \forall\, p \in [0,1]$
7: **if** $(p \leq rc)$ and $rapidly\_changing$ is not empty
8: pick a page that is in the $rapidly\_changing$ list and visit the page
9: **else if** $(p > rc \text{ and } p \leq mi)$ and $monitoring\_information$ list is not empty
10: pick a page that is in the $monitoring\_information$ list and visit the page
11: **else if** $(p \geq el)$ and $everything\_else$ list not empty
12: pick a page that is in the $everything\_else$ list and visit the page
13: **else if** at least one list is not empty
14: randomly select a link and visit the page from any of the three links
15: **else**
16: visit l
17: **end if**

Notice that there are some steps missing. Before we explain the missing steps, we will explain how we set some of the parameters. Based on the information in [42], we set $rc = 0.45$ (see table 4 for the definition of rc), meaning that 45% of the time that a page is being revisited, it will be from the $rapidly\_changing$ list. Recall that this list contains links to pages that users revisit in order to obtain new information. This makes the total probability of visiting pages in this list equal to $0.45 \times 0.5 = 0.225$. Also recall that the second list $monitoring\_information$ is a list that contains pages that users would generally go to for monitoring change. As per the results in [42], we set $mi = 0.75$ (see table 4 for definition of mi). This leaves every other site under the $everything\_else$ list.

5.3.5.2.4 Algorithm 2, Selecting Links to Add to Lists, and User Browsing Habits

The missing information relates to how lists acquire links which are selected (as in Algorithm 2.1) and added to them. We also describe how we identify links that fall under the three categories. Finally, we discuss how we set the size limit on each URL.

In order to mimic the behavior of an actual user, we researched users browsing habits and attempted to find browsing categories that can fall under one of the three lists in Algorithm 2.2.

The work done in [43] conducts a study on user behavior based on search and toolbar logs. They propose a three top-level CCS taxonomy of online page views. CCS stands for Content (news, portals, games, verticals, multimedia), Communications (email, social networking, forums, blogs, chat) and Search (Web search, item search, multimedia search). Table 5 shows their findings of user behavior on the Web based on the CCS taxonomy that they proposed.

Main category	Sub-category	Fraction
Content	Games	6.2
	Multimedia	5.4
	Portal	5.4
	Head Listings	3.4
	News	3.4
	Other Vertical	28.1
	Total	**52.0**
Communication	Social	24.3
	Mail	9.4
	Forum	1.4
	Blog	0.4
	Travel	**35.5**
Search	Main Search	6.2
	Multimedia Search	1.4
	Item Search	1.4
	Total	**9.0**
Unknown	Total	**3.4**

Table 5: Users Web browsing behavior habits by percentage [43]

Based on the table above on page visitation categories, content-based sites are occupying 52% of the webpages that users visit, 35% communication, 9% search and 3.4% unknown. In content, *Other Vertical* includes categories such as: Unknown (6.2%), Retail (3.4%), Travel (1.8%), Finance (1.4%), Sports (0.8%), Weather (0.2%) etc. Using the categories from page visit behavior in [43], we empirically mapped the categories in [43] to either $rapidly\_changing$ or $monitoring\_information$ lists. To be able to filter sites that fell under these categories, we empirically identified keywords that map to the categories in each list. In order to identify if a link falls under a certain list, we pass it through filters that check the URL for the keywords we have associated with them. In this manner, when a link l is not selected for an immediate visit on line 4 of Algorithm 2.2, it will be passed through the two filters. It first passes through the filter that identifies if a link should be placed in $rapidly\_changing$, then if not in that category, the same is done for

$monitoring\_information$. If it fails to be in either filter, it is then passed to the $everything\_else$ list. The table below shows which content we mapped with each list, and also a list of keywords we identified for them.

List	Size	Content	Sample Keywords
$rapidly\_changing$	$s_1 = 26$	News and Newspaper (Including Technology etc.), Finance, Weather, Sports, Blogs	news, sports, Barcelona, hockey, nytimes, usatoday, blog, report
$monitoring\_information$ (Including sites that might be visited more often than other sites)	$s_2 = 14$	Games, Entertainment, Movies, TV, Head listings (Ebay, Amazon etc.), Fashion	Amazon, mtv, hbo, imdb, rottentomatoes, kijiji, craigslist, comedy, concert
$everything\_else$	$s_3 = 13$	Every other category (e.g. Food, Personals etc.)	

Table 6: Size of lists

Ideally, multimedia sites such as *YouTube*, *Hulu* etc. could have been added to the list of $rapidly\_chanding$ sites. However, as these sites consume more bandwidth, we did not include them in the list. The sizes of the lists were based on the numbers in [42], where 19/39 URLs were for rapidly changing content etc. Proportionally to the findings in [42], we increased the size of the lists, since the numbers are not the number of sites users revisited. The sizes can be parameterized in a future work. To avoid the lists being filled up quickly, once the $rapidly\_chanding$ list reaches 10, future links that fall under this category are added to the list with a probability p_2. The value set for monitoring information is 6 (roughly half).

To increase the chances that a link selected will change for the two lists $rapidly\_changing$ and $monitoring\_information$, we find the host of link l and add

the host instead of the link returned by the search engine. For example, for $l = 'http://people.scs.carleton.ca/\sim ssajjadp/experimentPhaseTwo.html'$, the host in this case would be http://people.scs.carleton.ca/. In this manner, if the link selected relates to an article on a news sites, we don't take the actual article, but the homepage of the news site.

Before we present the actual Algorithm, table 6 describes the parameters involved in the algorithm, and the values we chose to set for our experiments.

Algorithm 2 ProfileMeNot

1: **when** query q is searched by Bing
2: filter and extract links returned using TrackMeNot and ProfileMeNot filters
3: randomly pick a link l from the extracted and filtered returned links from Bing
3: randomize $P, \forall P \in [0,1]$
4: **if** ($P \leq P_{revisit}$)
5: visit l
6: **else**
7: **if** ((l is a $rapidly\_changing$ page) AND (current size of $rapidly\_changing \leq s_1$))
8: randomize $p_{add\ link}, \forall p_{add\ link} \in [0,1]$
9: **if** ((current size of $rapidly\_chanding \leq l_1$) OR ($p_{add\ link} \leq p_1$))
10: parse host of l, h
11: **if** (h is not already in $rapidly\_changing$)
12: add host of l to $rapidly\_changing$
13: **end if**
14: **end if**
15: **else if** (l is a $monitoring\_information$ page) AND (size of $monitoring\_information$ list $\leq s_2$))
16: randomize $p_{add\ link}, \forall p_{add\ link} \in [0,1]$
17: **if** ((size of $monitoring\_information$ list $\leq l_2$) OR ($p_{add\ link} \leq p_2$))
18: parse host of l, h
19: **if** (h is not already in $monitoring\_information$ list)
20: add host of l to $monitoring\_information$ list
21: **end if**
22: **end if**
23: **else if** (size of $everything\_else$ list $\leq s_3$)
24: add l to $everything\_else$ list
25: **end if**
26: randomize $p, \forall p \in [0,1]$
27: **if** ($p \leq rc$) and $rapidly\_changing$ list is not empty
28: pick a random page that is in the $rapidly\_changing$ list and visit the page
29: **else if** ($p > rc\ and\ p \leq mi$) and $monitoring\_information$ list is not empty
30: pick a page that is in the $monitoring\_information$ list and visit the page
31: **else if** ($p \geq el$) and $everything\_else$ list not empty
32: pick a page that is in the $everything\_else$ list and visit the page
33: **else if** from either list $rapidly\_changing$, $monitoring\_information$ or $everything\_else$
34: randomly select a link and visit the page from any of the three links
35: **else**
36: visit l
37: **end if**
38: **end if**
39: **end when**

In order to simulate a user's behavior more accurately, we researched the average number of Web pages users view in a given time frame. We will discuss more on the values we selected for experimentation in chapter 6.

5.3.6 Strengths and Weaknesses of ProfileMeNot

ProfileMeNot, similar to the tool it was built on top of, TrackMeNot, is a client-side solution that requires no server side modifications and does not require trust in any third-party. Furthermore, it is easily installed and does not require complicated configurations. In Chapter 4 we compared ProfileMeNot with several other solutions. Now that we have examined in depth how ProfileMeNot functions, we will give a brief summary of the advantages of ProfileMeNot over other solutions. We will then describe the potential weaknesses of ProfileMeNot and discuss how they can be solved.

5.3.6.1 Strengths

The following five points are ProfileMeNot's strengths when compared to proposed and existing solutions.

- Compared to blocking tools, both partial and complete, ProfileMeNot does not need to engage in an arms race with trackers. As we saw in Chapter 3, trackers look for innovative ways to track users. Furthermore, most blocking tools are not effective against fingerprinting. NoScript, implemented as a Firefox extension, is a tool that is effective against fingerprinting. However, unlike ProfileMeNot, NoScript blocks advertisements. NoScript also only allows scripts from whitelisted sites to operate, creating a heavy toll on user experience. Moreover, ProfileMeNot does not need to keep an updated list of blacklisted domains. We also examined the private browsing mode of major browsers. In section 4.3.3, we

concluded that private browsing modes of different browsers have vulnerabilities and do not necessarily block tracking. ProfileMeNot is also very easily configured.

- In section 4.1 we saw that self-regulation and Do No Track are not widely adopted and do not necessarily block tracking. In this respect, ProfileMeNot does not need to place trust in any party. It also does not need to wait for any party to adapt to it.
- In section 4.2, we examined P3P and the case study on levis.com. As we concluded, users cannot trust a website's privacy policy, specifically with regards to third-party tracking. ProfileMeNot does not require any trust to be put in any website.
- As we saw in section 4.4, proxy-based anonymizers depend on a third-party exit-node. Users must put some degree of trust in at least one third-party. ProfileMeNot does not require any trust in a third-party.
- In section 4.5, we introduced possible solutions that allow for both privacy and OBA. We believe these classes of solutions are comprehensive and can replace the current methods employed in serving targeted ads. However, they have not been adopted and also do not claim or guarantee that smaller companies will not track users. We believe that if ProfileMeNot is widely used, third-party tracking companies and online advertisers will have more of an incentive to adopt architectures such as Adnostic [1] or RePriv [60]. In this regards, ProfileMeNot can be seen as an interim solution until the wider problem of tracking through both law and technology is solved.

- If users do not trust the developers of ProfileMeNot, given that ProfileMeNot is implemented as a Firefox extension, they can have the code verified by any entity they trust.

5.3.6.2 Potential Weaknesses

On the other end of the spectrum, ProfileMeNot does have some disadvantages. ProfileMeNot will skew data that Web analytic companies gather on users. It will also cause financial damage for advertisers who pay to have their ads viewed on a pay-per impression model. However, non-behaviorally-targeted ads that are paid per impression are generally very cheap (as low as 10 cents per 1000 impressions). We hope that ProfileMeNot will encourage the online advertising industry to adopt an Adnostic [1] or RePriv [60] type of architecture. ProfileMeNot does however still allow publishers to display their ads and hence be able to continue to provide free content. Additionally, our solution does not affect the pay per click advertising model.

An issue that users might find with ProfileMeNot is bandwidth consumption. We ran ProfileMeNot with a frequency rate of 60 search queries per hour and 120 search queries per hour using both algorithms for 48 hours straight. The results are demonstrated in Figure 7.

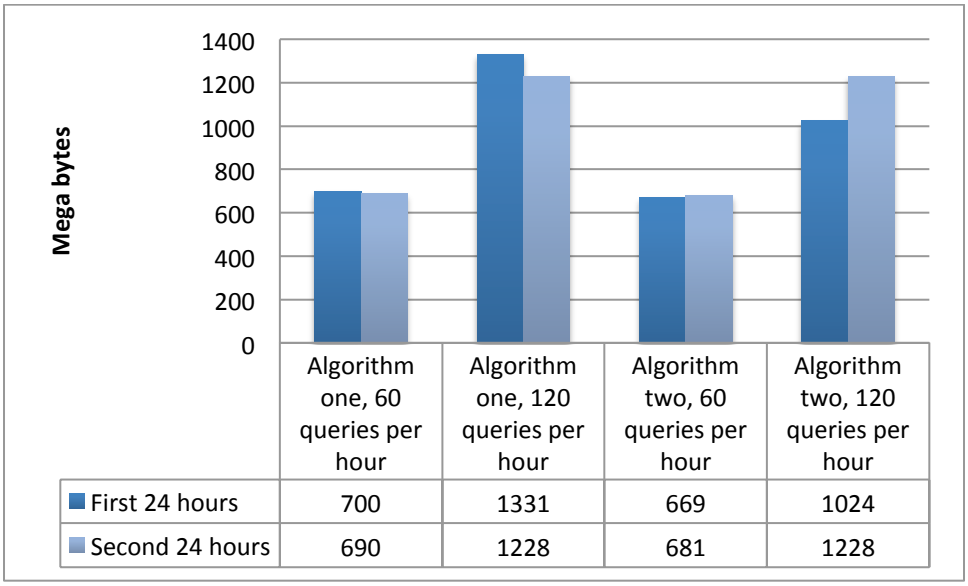


Figure 7: Amount of bandwidth used by ProfileMeNot in megabytes

On average, our volunteers browsed the Web using ProfileMeNot around 4.3 hours per day. The most consumed bandwidth was 1331 MB in 24 hours. Breaking it down into per hour, we get 55 MB per hour, which would mean $55 \times 4.3 = 236.5\, MB$ per day, and around 7 GB per month. However, this is an upper bound. As ProfileMeNot still opens some audio and video files, this number can be further reduced. In a survey we conducted prior to our experiments on 14 of the volunteers on how much bandwidth they are willing to allow ProfileMeNot to use in a span of 3 or 4 days, 28.57% respondents said 1 GB, 35.71% said 2GB, 14.29% said 5GB and 21.43% said more than 5 GB. Given the statistics, we do not believe that ProfileMeNot's bandwidth consumption will be an issue that will hold back users from installing it.

As ProfileMeNot is not currently designed to be very user-friendly, it occasionally opens some audio and video files. There are ways to stop audio and video files and we did inform our volunteers about them. However they require some amount of work, which most users will not do. In the future, we will incorporate such a function in our

design. Another issue with the current version of ProfileMeNot is that it opens the news links in a tabbed browser in the foreground by opening a new tab. We did provide users with instructions on how to hide the tabs that ProfileMeNot opens. This feature will be added to later versions of ProfileMeNot so that users do not have to manually hide the tabs.

To summarize, we believe that ProfileMeNot's advantages in addressing the privacy concerns of users supersedes its disadvantages. Additionally, most of ProfileMeNot's current drawbacks can be enhanced in future versions.

5.4 Conclusion

In this chapter we introduced and defined the concept of obfuscation. We gave examples of obfuscation in different domains. We further described how TrackMeNot has previously applied obfuscation in the context of Web search. We further explored how we built on top of TrackMeNot to use the technique of obfuscation in creating noise in a user's Web browsing history. We then introduced our two algorithms that ProfileMeNot utilizes to create noise in the behavioral profile of a user. We investigated the advantages and disadvantages of ProfileMeNot and concluded that its advantages outweigh its disadvantages.

6 Experiments, Results and Analysis

In this chapter, we firstly describe how we set the frequencies in our two algorithms. We then describe how we selected our volunteers and explain the taxonomy of Google's ads preferences. In section 6.2, we explain in detail how we conducted our experiment to verify the amount of noise introduced by using ProfileMeNot. We then present the results and analyze them.

6.1 Experiment Preliminaries

In this section, prior to explaining the methodology of our experiment, we will explain the frequencies we used for the number of search queries and link visits per hour done by ProfileMeNot. After this step, we explain the different steps of the experiment and explain how we selected our volunteers.

6.1.1 Frequency Rates

Kumar el al. [43] found statistics on how much time users spent online and the number of pages they viewed. Their key findings with regards to page views and time spent online include:

- The median number of page views per day is 59 and the median time spent online is around 1 hour per day
- In an individual session, the median length of page views is 17 page views in 16 minutes.

Given that our volunteers browsed the Web differently, we based the frequency of sites visited in our experiments on the median user's visitation habits. From both the bullet pointers above, it can be concluded that the median number of pages visited during a browsing session is approximately one page visit per minute. However, given that this is

the median, we decided to experiment with twice the rate and half the rate. However, due to the limited number of volunteers we had, we were only able to experiment with two link visit frequency rates.

For Algorithm 1, we tested two Web search query rates. The first rate is an average of 120 search queries per hour, which means a new link is visited twice every minute. The other frequency we set was an average of 60 search queries per hour, which means a new link is visited twice every two minutes.

For algorithm 2, we set the same rates. The algorithm, when the frequency is set to 120 search queries per hour, visits a new link every 30 seconds and every other 30 seconds it visits a previously selected link. When the frequency is set to 60 search queries per hour, it visits a new link every minute and a previously selected link every other minute.

As is evident, when the frequency is set to 120 search queries per hour, both algorithms visit two links every minute. This is twice the median page view of an average user as found in [43]. The other frequency visits a link every 1 minute, which is equal to the median page view in [43].

6.1.2 Volunteer Selection and breakdown

We found around 28 volunteers to help us measure the effectiveness of ProfileMeNot. Given our limited resources and the amount of work and trust required for the experiment, we were not able to find any additional volunteers. Three of our volunteers also dropped out in the middle of the experiment. Luckily, two of our volunteers agreed to help us with both algorithms. We also had two additional volunteers join during the experiment process. However, given that the volunteers joined in and dropped out in at different times, we ended up having eight volunteers for two of our test sets, seven for

another, and six for the remaining experiment. We tried to get a diverse audience from different genders, cultures, language background, ages, and social status. The following table summarizes the basic demographics of our volunteers. Due to privacy concerns, we do not specify the number of each category.

Country of residence	Canada, United States, UK, UAE, Mexico
Language	English, French, Persian, German, Spanish, Arabic, Guajarati, Urdu
Age group	18-24, 25-34, 35-44
Gender	Female, Male

Table 4: Demographics of volunteers

After selecting our volunteers, we divided them into four groups. We tested both algorithms in the two aforementioned time frequencies. The four groups were:

1) Algorithm 1 with an average frequency of 120 search queries per hour.
2) Algorithm 1 with an average frequency of 60 search queries per hour.
3) Algorithm 2 with an average frequency of 120 search queries per hour.
4) Algorithm 2 with an average frequency of 60 search queries per hour.

Before we explain the different steps we took to conduct our experiments, we explain Google's ads preferences taxonomy.

6.1.3 Google Ads Preferences and Taxonomy

As of 2008, Google AdSense contains 91,000 unique domains that it uses to track users [1]. As we saw in Chapter 4, Google builds a behavioral profile of each user and allows them to view and edit it. Google's ads preferences taxonomy includes four layers. The first layer is a general layer. We use the same terminology employed in [17] to identify the different layers. We call the categories in the first layer *roots*. There are currently 25 root categories. The second layer is a sub-category of the root. We call elements in the

second layer *parents*. Each root has several parents. The third layer is a sub-category of the parent. We call elements in this layer *nodes*. Some parent categories contain nodes, and some do not. The fourth layer is for a very limited number of categories. For example in *Online Communities→ Dating & Personals → Photo Rating Sites*, *Online Communities* is the root, *Dating & Personals* is the parent, and *Photo Rating Sites* is the node. For the full list of Google roots as of March 2013, see appendix A.

As Google states, it builds a profile for users.

"Many websites, such as news sites and blogs, partner with us to show ads to their visitors. To see ads that are more related to you and your interests, edit the categories below, which are based on sites you have recently visited." "Your interests are associated with an advertising cookie that's stored in your browser. If you don't want us to store your interests, you can opt out below. Your ads preferences only apply in this browser on this computer. They are reset if you delete your browser's cookies."

Given that Google and its affiliates are currently the largest ad-network and that their behavioral profile of users can be viewed and edited, we set the changes in Google Ads preferences as our means of quantifying the behavior of ProfileMeNot. We also asked users to provide us with Yahoo-Bing ads manager. After initial empirical analysis, we found Google's ads preferences to be more accurate than Yahoo-Bing and set out our experiments on the profiles built by Google.

6.1.4 Experiment Steps

Our experiment composed of four major steps. Keeping in mind the bigger picture, we asked volunteers to send their current Google's ads preferences, install and use Firefox,

opt-in to Google ads preferences, send their ads preferences after a few days of using Firefox, install ProfileMeNot, use ProfileMeNot for a few days, send their ads preference, and again use ProfileMeNot and send their ads preferences. The detailed explanations of the steps are as follows:

- In a detailed set of instructions, we asked volunteers to send their Google ads preferences from the current browser they were using. In the same set of instructions, we asked them to install Firefox and opt-in to Google ads preferences. In order for the behavioral profile of users to be built on Firefox, we asked them to use Firefox for four to six days. Fortunately, the average number of days used for building their profile was extended to 11 days higher number of days using Firefox will build a more accurate Google ads preferences profile. We created a FAQ and a video to assist them with the instructions. The instructions of this step can be found in appendix B.1.
- After they had used Firefox for around 11 days, we sent our volunteers ProfileMeNot as a Firefox add-on. Along with the add-on, we also sent a PDF file with instructions on how to install the add-on. We created a site for this phase of the experiment to keep the volunteers updated with any issues they might have. The site was updated frequently, addressing commonly faced issues. For example, we provided instructions on how to block audio files contained in links from playing automatically. A video was also made to further assist the volunteers with the tasks. We also asked the volunteers to send us the ads preferences built on them by Google from their Firefox. The PDF instructions can be found in appendix B.2.

- Around 3-4 days after our volunteers used ProfileMeNot, we sent a follow-up email asking our volunteers to send us their Google ads preferences profile. Most volunteers sent us their profile within five to eight days. We also asked them to send us the log file that ProfileMeNot generates. The log file includes the list of search queries and link visits that ProfileMeNot performed.
- In another follow up email, after having received their profiles in step 3, we asked users to send us their profile and logs again. In this step, we asked volunteers to identify the interest categories in their profile which they think are not related to websites they have visited.

We asked the volunteers who were given Algorithm 2 to use, if possible, the add-on for a few extra days. Fortunately some of them complied. However, coincidently, the volunteers that complied were volunteers that sent their results in steps three and four on time. Therefore, the average number of days using ProfileMeNot increased. We would have conducted the experiment for more days, but most users were reluctant to use Firefox and ProfileMeNot. Figures 8, 9, 10, and 11 show the average number of days each algorithm with a given frequency was used.

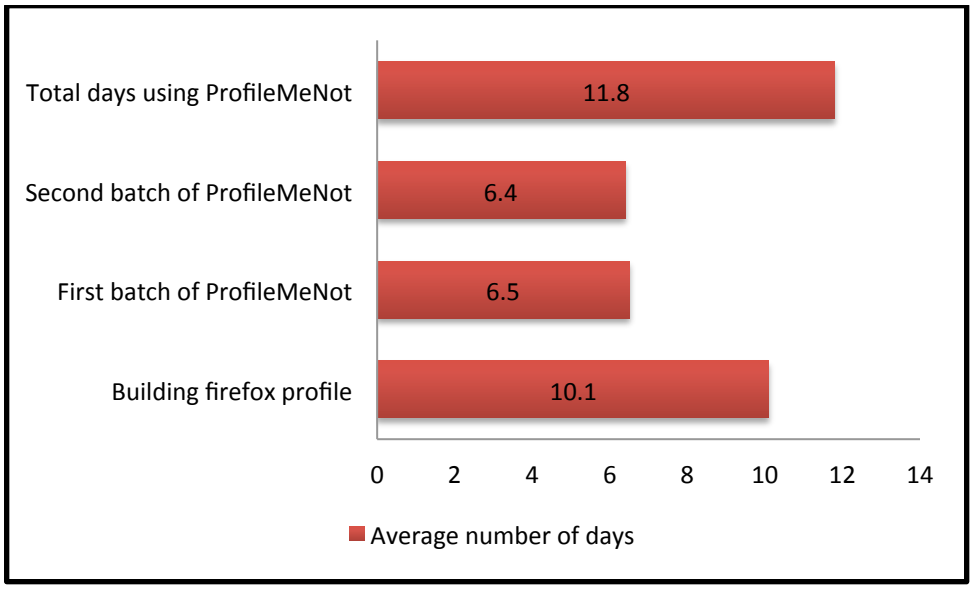


Figure 8: Average number of days ProfileMeNot was used by volunteers in Algorithm 1 with a frequency of 120 queries per hour

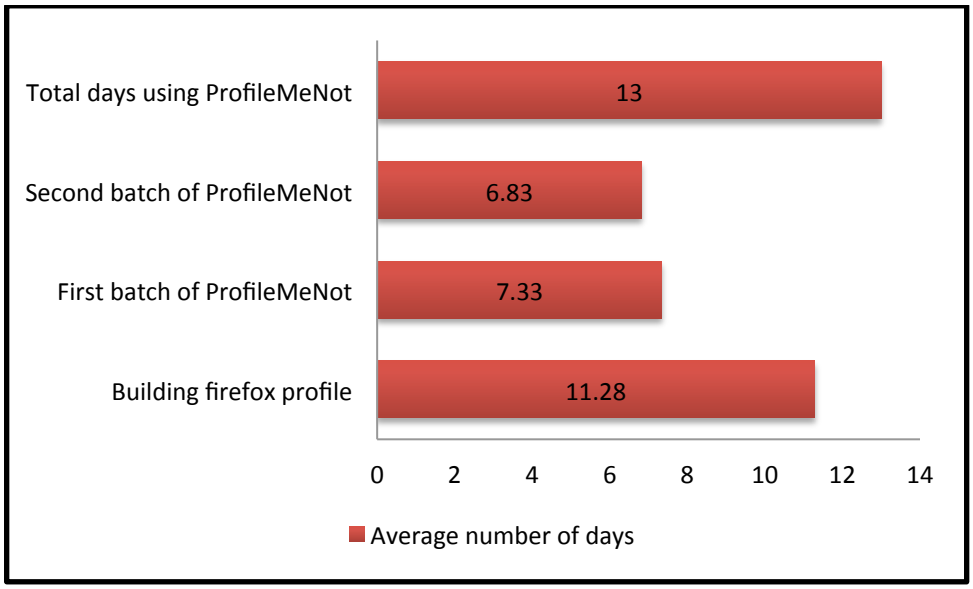


Figure 9: Average number of days ProfileMeNot was used by volunteers in Algorithm 1 with a frequency of 60 queries per hour

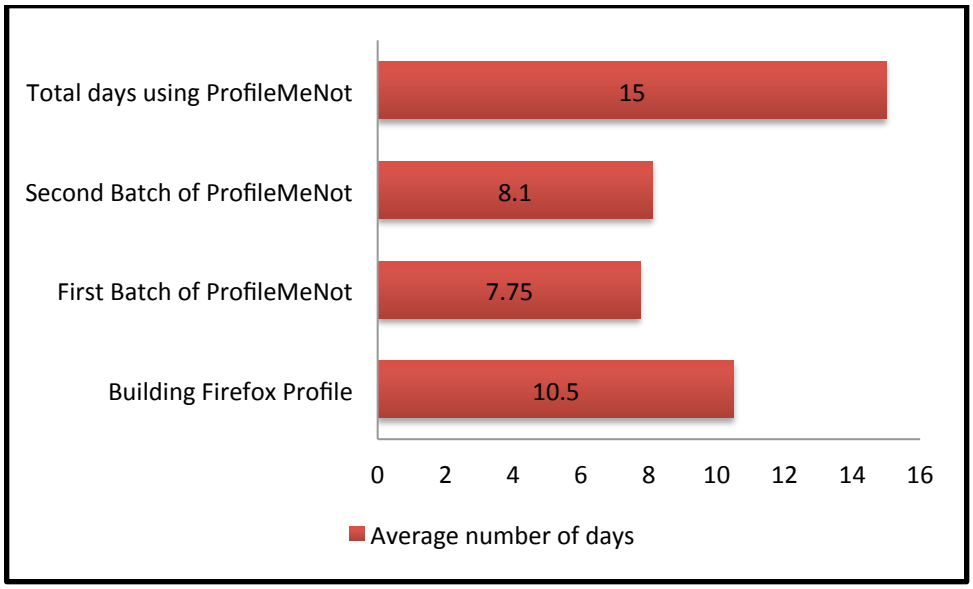


Figure 10: Average number of days ProfileMeNot was used by volunteers in Algorithm 2 with a frequency of 120 queries per hour

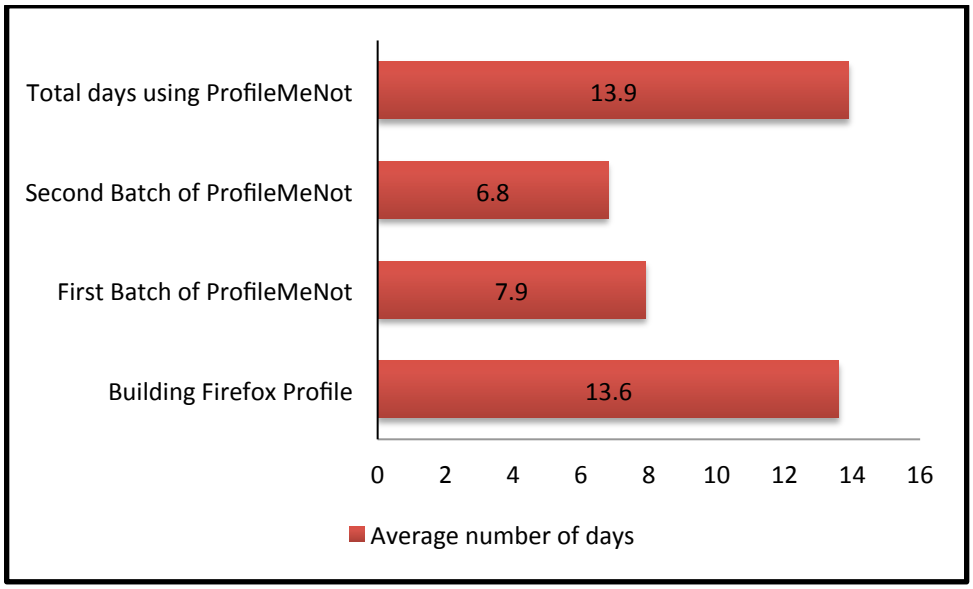


Figure 11: Average number of days ProfileMeNot was used by volunteers Algorithm 2 with a frequency of 60 queries per hour

6.2 Comparison Methodology and Results

In this section, we describe how we quantify the amount of noise introduced by using ProfileMeNot. We will do this by explaining our comparison methodology.

6.2.1 Comparison Methodology

To measure the amount of noise ProfileMeNot introduced, we compared each volunteer's profile and examined the differences between the volunteers' interests and those that were built due to ProfileMeNot. To ensure that we correctly distinguish between a volunteer's actual interests and the fake interests produced, we used the following information:

1) The volunteer's Google ads preferences profile before using Firefox.
2) The volunteer's Google ads preferences profile after using Firefox for a few days.
3) The interests that a volunteer had indicated were not theirs from step four of the experiment. Therefore, the remaining become the volunteers interests.

If any category belonged to any of the above three, we placed it as volunteer's interest. We noticed that some categories disappear each time a user sends us their profile. We concluded that this is due to the fact that either a user hasn't recently visited a page that is in that category that disappeared, or ProfileMeNot did not visit such a category. Therefore, any category that disappears is not considered in our results, as we are concerned with the profile built on a user at any given time. Volunteers who had 2 or fewer root categories and 3 or fewer parent categories in both the profiles sent from their default browser and after using Firefox for around 10 days, were taken off the general statistics as this shows that they did not frequently visit websites that are in Google's

advertising network. We compared each volunteer's results on three levels. Our work in this section is similar to the category comparison method in [17].

1) Root level: We used the three aforementioned datasets we had available to distinguish between volunteers' actual interests and the fake interests inferred. Each root category that was not in either set was considered and created by ProfileMeNot. We only counted each root category once, as the redundant uses of the same root category are as results of parent and node layers. Those layers are compared separately.
2) Parent level: For each root category, we compared its parents with the three datasets in a similar fashion to the root level.
3) Node level: We conducted the same methodology as the parent level to compare results. Note that in each layer, we only looked at information in that layer.

For each time a volunteer had sent us a profile while using ProfileMeNot, we conducted the three-tier comparisons. For every layer, we counted the number of actual volunteer categories and the fake categories added. We then calculated the amount of noise introduced by calculating the percentage of noise created based on the number of actual volunteer interests. For example, if the actual volunteer root categories were five, and ten additional categories were added, the percent of noise introduced would then be (10/5) * 100 = 200%. This would mean that a profiler would be able to guess if a root category belongs to a user with a probability of 1/3.

For all four groups, we conducted the above comparisons with respect to the following.

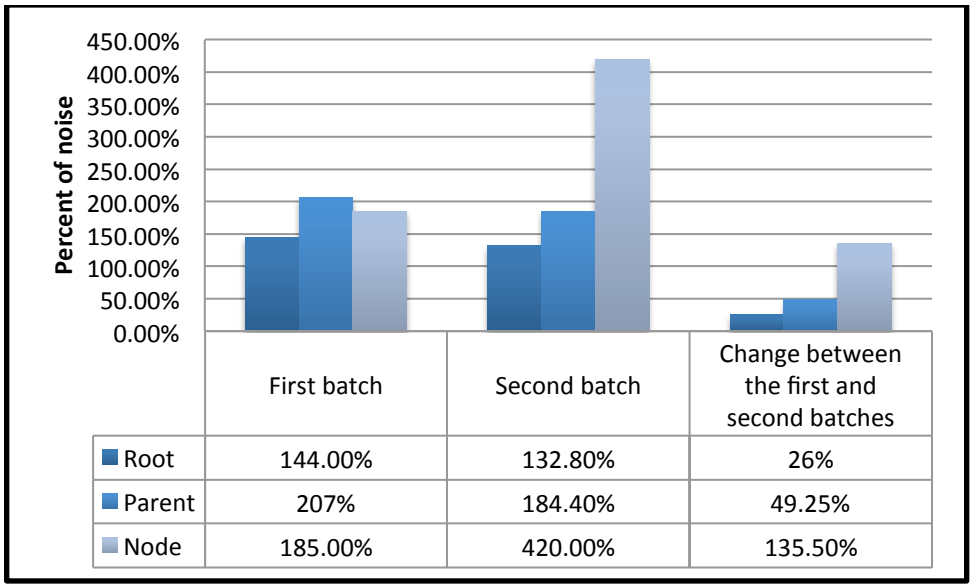
1) The first profile they sent us after using ProfileMeNot.
2) The second profile they sent us after using ProfileMeNot.

3) The difference between the first and second profiles sent while using ProfileMeNot. The purpose of this comparison is to see how much new information has been added to a volunteer's profile after using ProfileMeNot for an additional few days. In this comparison, we linked any category (comparing in all three levels) produced in the first profile using ProfileMeNot and any of the volunteer's actual interest categories to newly introduced categories in each layer.

We also calculated the difference in noise created in each batch by both algorithms in the same frequency. We did this by subtracting the percentages of change in Algorithm 2 from Algorithm 1. For example, if Algorithm 1 introduced 100% noise in the root level when the frequency is set to 120 queries per minute and Algorithm 2 introduced 75% noise for the same settings, the difference would then be 100% - 75% = 25% difference.

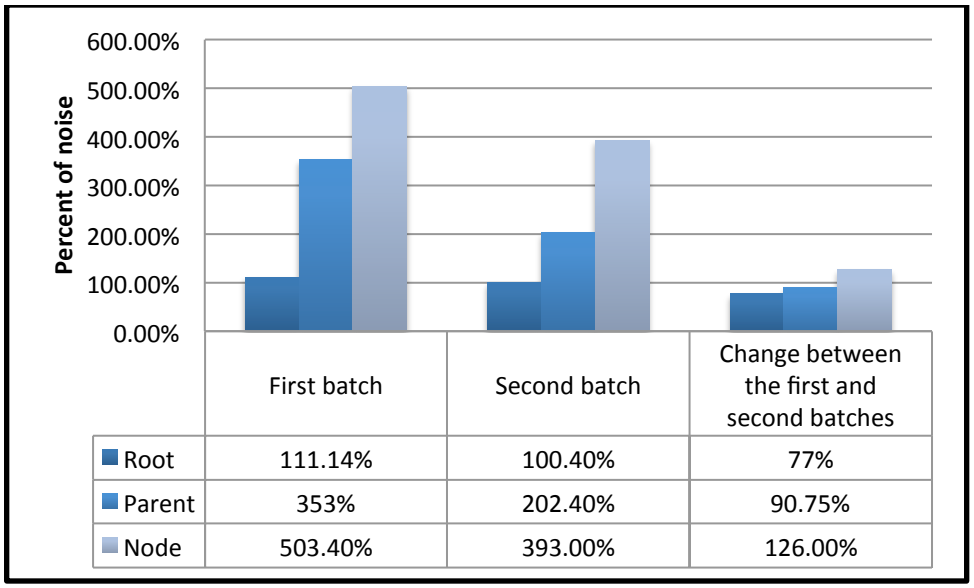
6.3 Results

Figures 12, 13, 14, and 15 demonstrate the average results of each group of volunteers. From left to right, each figure contains the percent of noise introduced in the three different layers from comparing the three points mentioned at the end of section 6.2.1.



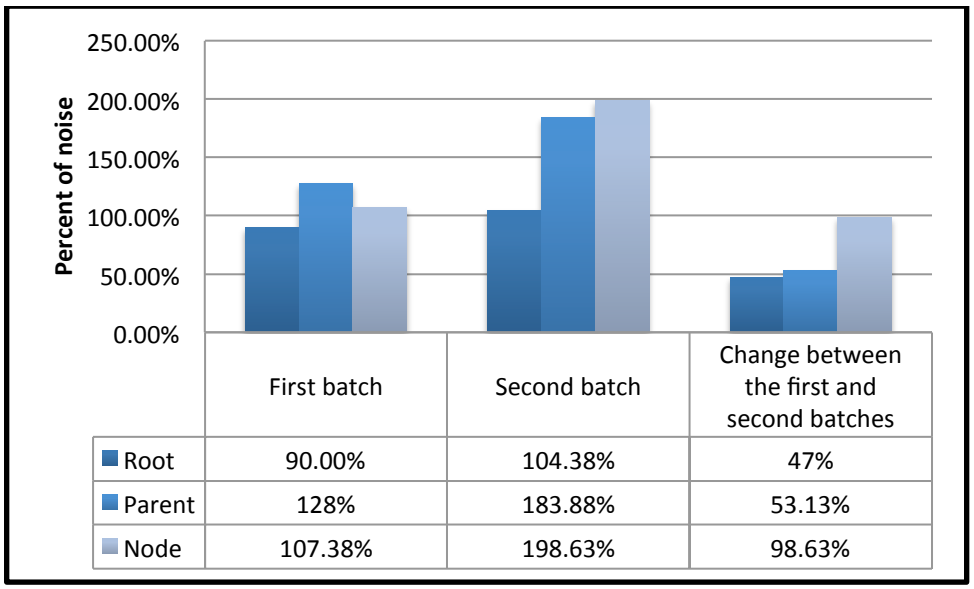
	First batch	Second batch	Change between the first and second batches
Root	144.00%	132.80%	26%
Parent	207%	184.40%	49.25%
Node	185.00%	420.00%	135.50%

Figure 12: Average percent of noise in Algorithm 1 with an average frequency rate of 120 queries per hour



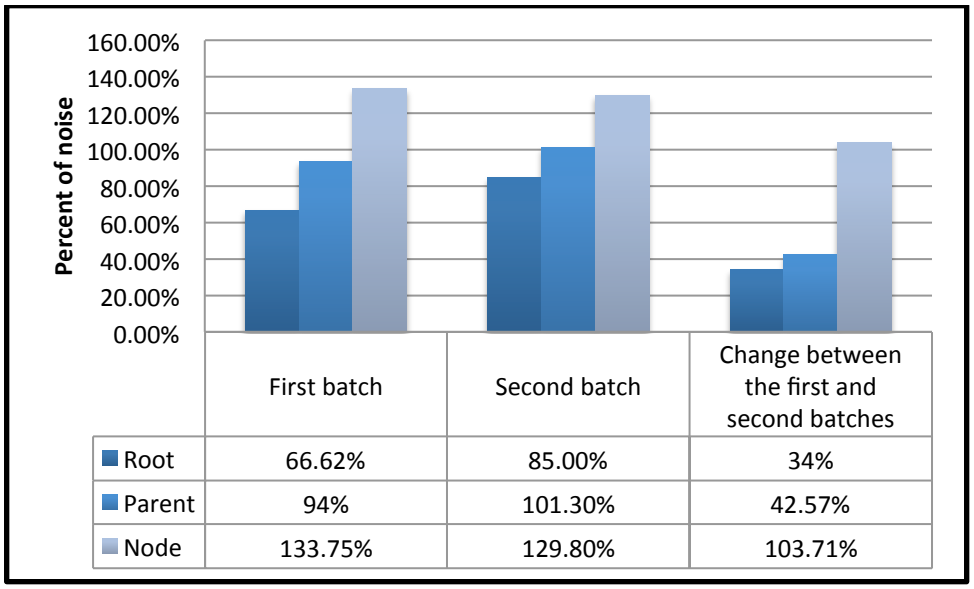
	First batch	Second batch	Change between the first and second batches
Root	111.14%	100.40%	77%
Parent	353%	202.40%	90.75%
Node	503.40%	393.00%	126.00%

Figure 13: Average percent of noise in Algorithm 1 with an average frequency rate of 60 queries per hour



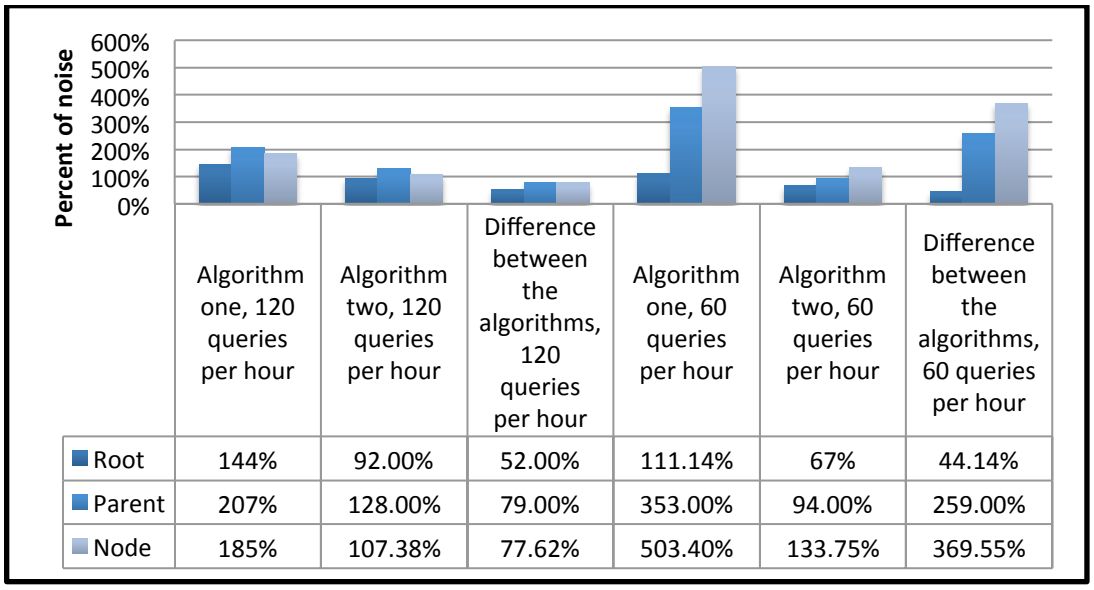
	First batch	Second batch	Change between the first and second batches
Root	90.00%	104.38%	47%
Parent	128%	183.88%	53.13%
Node	107.38%	198.63%	98.63%

Figure 14: Average percent of noise in c Algorithm 2 with average frequency rate of 120 queries per hour



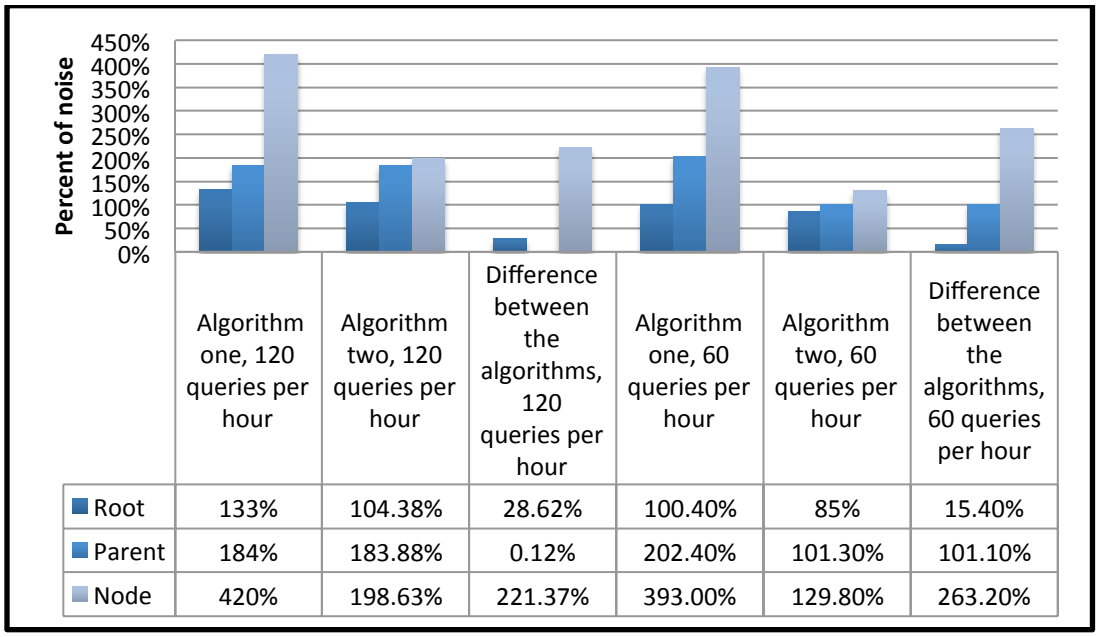
	First batch	Second batch	Change between the first and second batches
Root	66.62%	85.00%	34%
Parent	94%	101.30%	42.57%
Node	133.75%	129.80%	103.71%

Figure 15: Average percent of noise in Algorithm 2 with average frequency rate of 60 queries per minute



	Algorithm one, 120 queries per hour	Algorithm two, 120 queries per hour	Difference between the algorithms, 120 queries per hour	Algorithm one, 60 queries per hour	Algorithm two, 60 queries per hour	Difference between the algorithms, 60 queries per hour
Root	144%	92.00%	52.00%	111.14%	67%	44.14%
Parent	207%	128.00%	79.00%	353.00%	94.00%	259.00%
Node	185%	107.38%	77.62%	503.40%	133.75%	369.55%

Figure 16: Comparison of algorithms with different frequencies in the first batch



	Algorithm one, 120 queries per hour	Algorithm two, 120 queries per hour	Difference between the algorithms, 120 queries per hour	Algorithm one, 60 queries per hour	Algorithm two, 60 queries per hour	Difference between the algorithms, 60 queries per hour
Root	133%	104.38%	28.62%	100.40%	85%	15.40%
Parent	184%	183.88%	0.12%	202.40%	101.30%	101.10%
Node	420%	198.63%	221.37%	393.00%	129.80%	263.20%

Figure 17: Comparison of algorithms with different frequencies in the second batch. The third and sixth columns show the difference in percentage change of each.

In the next few sub-sections, we first discuss the results of Algorithm 1 and Algorithm 2 separately. We then discuss the commonalities of results in both and move on to discuss their differences. Lastly, we describe the limitations of our results.

6.3.1 Algorithm 1 Results

In Algorithm 1, as Figure 12 demonstrates, we can see that the amount of noise introduced is around 130-140% in the root category when the frequency is set to 120 queries per hour, and between 100-111% when the frequency is set to 60 queries per hour. This shows that changing the frequency has a noticeable effect in the noise introduced in the root level (around 40%). In the first batch, the lower frequency has a higher noise introduced in the parent and node level. It is difficult to determine why exactly this is the case. There are a few possible explanations. Given that the number of volunteers we had was limited, we noticed that two of the volunteers in the lower frequency of Algorithm 1 had introduced more noise than the others. This could be due to the fact that they have visited fewer sites that Google ads preferences picked up as their

parent level interest. Another explanation could be that by chance, since there are randomizations involved, the queries that were selected for users in this category were more closely related. Hence many sites were visited within the same root category. However, as Figure 13 demonstrates, in the second batch, we see that that the node layer of the higher frequency has introduced around 30% more noise than the lesser frequency. Generally, the root level indicates a user's core interests. In this respect, we believe that Algorithm 1 successfully obfuscates the actual interests of a user in a pool of fake interests. However, as our goal was to keep online behavioral advertising alive and to also mimic an actual user's behavior, we developed Algorithm 2.

6.3.2 Algorithm 2 Results

With the exception of the change in nodes in the first batch, as evident in Figures 14 and 15, Algorithm 2 with a frequency of 120 queries introduces more noise then running with 60 queries per hour. The noise introduced in the root level of Algorithm 2's higher frequency ranges between 90-104% and 67%-85% in the lower frequency. The difference between the two is around 30-40%. In the second batch, we can see that the difference between the parent and node of the two are around 80% and 70% respectively. We believe that the second batch introduced a more accurate view of Algorithm 2, as it requires some time to build up the list of sites it needs to revisit. We can say that the higher frequency produces around twice as much noise as the lesser frequency in the parent and node level. In the root level, given that there are fewer options (25 root levels), it is unlikely that doubling or halving the number of link visits will have a linear effect on the change. The same argument can be made for parent and node levels; however given

that the options are more, the noise introduced has, in this case, a near linear correlation with the frequency.

6.3.3 Commonalities Between Both Algorithms

Figures 16 and 17 demonstrate the difference in change between the two algorithms. The change is calculated by subtracting the changes in Algorithm 1 categories from Algorithm 2 categories.

When we compare the two algorithms, we can see that both algorithms create around 100% or more noise. In this regard, both algorithms have the capacity to effectively obfuscate the interest categories of a user. Figures 16 and 17 also demonstrate that the root level changes are more in Algorithm 1. However, with the exception of the parent change in the second batch between Algorithm 1 and Algorithm 2 with a frequency of 120 queries per hour, we see that the root level changes are not substantially different. The reason behind this can be that the number of root categories is limited (25; see appendix A). The limited number of roots would mean that the sites ProfileMeNot visits have a higher chance of colliding with the interest categories of a user. We noted that many of the volunteers had *Arts & Entertainment* in their original root categories. When we asked volunteers to identify the interest categories that they thought were not theirs after having used ProfileMeNot, many chose some parent or node category under *Arts & Entertainment* while indicating that some other parent or node categories were part of their interests. In the second batch, we see that the percent of noise in the parent level for the frequency of 120 queries per hour is about the same for both algorithms. This can also be caused by the limited number of parent categories and the fact that all the users are using the same RSS feeds, hence visiting pages in a limited cluster of interests.

6.3.4 Differences Between Both Algorithms

As is evident in Figures 16 and 17, Algorithm 2 produces less noise than Algorithm 1 in both frequencies, both batches, and all three layers. Our work shows that the amount of noise introduced in a user's profile can be controlled.

However, as the node level is more diverse, it gives us a better indication of the difference between the two algorithms. Algorithm 1 produces around twice as much noise in the second batch in the node layer than Algorithm 1 in the higher frequency. In the lower frequency, Algorithm 1 produces around three times the noise. While these results are not linear, they do demonstrate the effectiveness of Algorithm 2 in introducing less noise while creating the same number of page views. This shows us that Algorithm 2 was effective in controlling the amount of noise introduced.

As we specified in Chapter 5, we enabled the burst-mode of TrackMeNot. This was in order to have visited a few websites in a row that are more closely related to each other. We believe that enabling this mode helped in controlling the amount of noise that was introduced.

As noted, the relation between the frequencies and algorithms in the different layers is not linear. A simple reason, as mentioned earlier, that the results are not linear can be due to the fact that the interests of volunteers collides with ProfileMeNot's interests, especially in the root categories as the root category is limited.

6.3.5 Expectations and effects on OBA

Our hypothesis was that Algorithm 1 would produce more noise than Algorithm 2. We did not have an estimate on an upper bound on the amount of noise introduced in Algorithm 1, but we aimed to introduce 100% noise, at least in the parent level, of

Algorithm 2. We tried to achieve this by investigating user revisitation and browsing habit patterns. We also looked at the number of page views users perform to set our frequencies. To reach our goal, we experimented with the median page visit frequency of users based on the results on [42]. We tested twice the frequency and half the frequency to see which frequency achieves our goal. As our results demonstrate, the second batch of results of Algorithm 2 with a frequency of 60 queries per minute achieves results close to our expectations. The first batch results are also not far off our expectations but the second batch is more accurate, as it needs time to build up its list of revisitation sites. In the second batch, this algorithm and frequency also introduce 85% and 129% noise in the root and node level, which is not far off from 100%. In this respect, as we saw in section 2.2.2, given the percent of targeted ads are around 4-5%, the expected decrease in accuracy of targeted ads while using ProfileMeNot drops to 2-3%. Therefore, we did not eliminate targeted advertising. Many of the proposed solutions completely kill targeted advertising. We do acknowledge that our solution will have an impact on the business of advertisers who pay per impression. However, the effects are less severe than blocking all ads as this will hurt both publishers and advertisers. Advertisers that pay per click do not get affected. Furthermore, pay per impression ads are generally very cheap and can be as low as 10 cents for 1000 impressions.

6.3.6 Accuracy of Results

We cannot claim that our results are completely replicable. If others conduct our experiment, the results may vary depending on the audience and the browsing habits of their volunteers. A challenge in our experiments is that we did not have control over what volunteers did and if they followed our steps correctly. We did try to provide users with

every possible means of assistance to get the experiment tasks done. We made videos, held Google hangout sessions and for a select few, set up everything on their computer. We were not able to extend the period of our experiment, as our volunteers were reluctant to continue using Firefox as their main browser. Additionally, we would have ideally liked to have had many more participants. However, given the nature of the work it was difficult to find more volunteers. However, we do believe that our results give relatively accurate results as changing the algorithms and frequencies met our expectations in terms of controlling the amount of noise introduced. Given the mentioned fact, we have developed a proof of concept that obfuscation as tool to achieve privacy in OBA works. Other researchers conducting our experiments might obtain different values; however, we empirically believe that the difference in the results of the Algorithms and frequencies will not be drastically different.

To the best of our knowledge, it is uncommon to quantify the amount of noise introduced with respect to third-party trackers. For example, TrackMeNot does not introduce a quantification methodology.

A possible alternative in quantifying the amount of noise introduced is to have a version of ProfileMeNot running along with a robot that simulates a user's behavior. ProfileMeNot should then infer the interest categories of the websites visited by both in order for it to distinguish between the robot's interests and its interests. The experiment can then be run on multiple Firefox profiles. We did not have enough resources to create a robot and implement or integrate a client-side interest categorization tool. In such a scenario, Google's ads preferences profile probably cannot be used, as ProfileMeNot may not be able to confidently map Google's interests' categories to the sites it maps.

6.4 Conclusion

In this chapter, we explained how we selected which frequency rates to use in order to conduct our experiments and how we selected our volunteers. We demonstrated how Google's ads preferences profile taxonomy is structured and specified that we quantify the changes in the behavioral profile of a user before and after using ProfileMeNot on Google's ads preferences. We further explained in detail how we conducted our experiments.

In the results section, we demonstrated that it is firstly possible to create noise in the behavioral profile of a user. Secondly, we demonstrated that it is possible to control the amount of noise introduced. Thirdly, we saw that Algorithm 2 with a frequency of 60 Web search queries per hour, can produce around 100% noise in all three layers of the Google ads preferences profile taxonomy. This would mean that a third-party tracker or any party that profiles the user via tracking them across websites can distinguish between a user's actual interests and fake interests with a probability of $1/2$. Furthermore, with the amount of noise introduced, the expected change in the accuracy of targeted advertisements should fall to ½. This would mean that instead of 4-5% of ads being targeted ads, 2-3% will successfully be targeted ads. We believe this will not drastically impact advertisers who pay per impression for their targeted ads. For advertisers who pay per click, their revenues remain unchanged. However, given the impact it will have on pay per impression advertisements, we hope that our solution will encourage advertisers and the industry to approach an Adnostic [1] or RePriv [60] type of solution.

7 Conclusion and Future Work

In our work we demonstrated the privacy concerns that arise from third-party tracking. We then investigated different techniques that trackers use to track users across websites in Chapter 3. We concluded, as in [34], that if the bigger picture of third-party tracking isn't solved, we risk an arms race with the trackers. In Chapter 4, we surveyed the different solutions that have been proposed or exist to solve the privacy concerns. We examined each solution's strengths and weaknesses. While examining each solution, we specified how ProfileMeNot overcomes the weaknesses of each solution. We concluded that current solutions do not protect the privacy of users adequately, are not widely adopted, or are not feasible to use.

In Chapter 5, we presented our algorithms. We demonstrated that obfuscation can change the behavioral profile of a user. Furthermore, with Algorithm 2, we proved that it is possible to control the amount of noise that is introduced in the behavioral profile of a user. Algorithm 2 with a frequency of 60 queries per hour managed to create 85% noise in the root layer, 101% noise in the parent layer and 129% noise in the node layer. The results of Algorithm 2 confirm that our hypothesis that simulating the behavior of a user, we can produce the same number of fake interest categories as the number of true interest categories for a user. This met the expectations that we had while designing Algorithm 2. Furthermore, as we saw in section 2.2.2, the percent of online-targeted ads is around 4-5%. ProfileMeNot will only decrease the accuracy of targeted ads by half; hence 2-3% of ads become correctly targeted. Moreover, although ProfileMeNot does impact the pay-per impression-advertising model, unlike other tools that totally block targeted advertising, ProfileMeNot allows for some advertising to occur. The negative business

impact that it has is on advertisers and not publisher websites that display the ads, hence having positive impact on publisher websites. Given this fact, we believe that ProfileMeNot will not affect free content on the Web. We hope that this will encourage advertisers and the industry to adopt an Adnostic or RePriv model.

7.1 Future work

Future work on these projects entails two paths. One is enhancing the current version of ProfileMeNot, and another is taking the path of selective obfuscation in lieu of general obfuscation. By general obfuscation we mean creating general noise instead noise tailored to a user.

With regards to changes to the current version, Algorithm 2 of ProfileMeNot does not change the list of sites it revisits. Due to time limitations, we did not research how often people change their browsing behaviors and how often they are likely to find a link to revisit. Even if we had done so, we did not have enough volunteer time to evaluate the results. Such sites would require evaluation over a period of months.

We do not see it necessary to change the parameters in Algorithm 2 as the results are satisfactory with a 60 query per hour frequency. However, the parameters can be changed with different query frequencies and evaluated.

As mentioned, ProfileMeNot currently employs a general obfuscation technique. The algorithms do not consider the user's browsing behavior while visiting links. ProfileMeNot acts intelligently when a user is performing Web searches through TrackMeNot's real time search awareness feature. However, both TrackMeNot and ProfileMeNot do not consider the behavior of the user. Further research can be directed toward a selective obfuscation model. In such a model, ProfileMeNot can infer the

interests of the users and, based on those interests, intelligently visit links. It can also keep track of interests that are faked and those that are real. This approach also helps limit the bandwidth usage of ProfileMeNot, as it will decrease the collision in the root, parent, or node category of websites that it visits and those that the user visits.

Furthermore, future obfuscation approaches do not have to be built on top of TrackMeNot. Our original intent was to take data from social bookmarking websites such as delicious.com and reddit.com. However, due to the technical difficulties that we faced, we decided to build on top of an existing solution. ProfileMeNot can also take links from Google trends in order to mimic a user's behavior more closely. Furthermore, ProfileMeNot can also block cookies from tracking companies that are trusted, so that no noise is created in their profile. By *trusted companies* we mean companies that have transparently proved that they address the privacy concerns of users. This can be in the form of a whitelist that a ProfileMeNot user can selectively add.

Aside from the obfuscation strategy, we believe that solutions such as Adnostic [1] or RePriv [60] should be the architectures that the industry should adopt. These architectures allow for both targeted advertising and privacy at the same time. However, the industry hasn't adopted either model. We do acknowledge that ProfileMeNot will skew Web analytics data and will also have an effect on advertisers that pay per impression. We hope that ProfileMeNot encourages the industry and in particular advertisers and advertising networks to approach either model, as they have not yet ensured users that their data will remain private.

List of References

Abbreviations used in this list:

SSRN: Social Science Research Network

LJ: Law Journal

ISJLP: A Journal of Law and Policy for the Information Society

[1] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas, "Adnostic: Privacy preserving targeted advertising," in *Proceedings of the 2010 Network and Distributed System Security Symposium,* 2010.

[2] J. R. Mayer and J. C. Mitchell, "Third-Party Web Tracking: Policy and Technology," in *2012 IEEE Symposium on Security and Privacy (SP)*, pp. 413–427, 2012.

[3] W. Luo, Q. Xie, and U. Hengartner, "Facecloak: An architecture for user privacy on social networking sites," in *Computational Science and Engineering, 2009. CSE'09. International Conference on*, vol. 3, pp. 26–33, 2009.

[4] "Targeted advertising," *Wikipedia, the free encyclopedia*. 13-Mar-2013.

[5] K. Li and T. C. Du, "Building a targeted mobile advertising system for location-based services," *Decis. Support Syst.*, vol. 54, no. 1, pp. 1–8, 2012.

[6] "Mobile App Tracking, Tracking Methods for Mobile Applications," *hasoffers*. [Online]. Available: http://www.mobileapptracking.com/docs/MAT-Tracking-Methods-For-Mobile-Apps.pdf. [Accessed 21-March-2013].

[7] H. Haddadi, P. Hui, and I. Brown, "MobiAd: private and scalable mobile advertising," in *Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture*, pp. 33–38, 2010.

[8] "Recommender system," *Wikipedia, the free encyclopedia*. 22-Mar-2013.

[9] J.-M. Dinant, C. Lazaro, Y. Poullet, N. Lefever, and A. Rouvroy, "Application of Convention 108 to the profiling mechanism," *Council of Europe*, vol. 35, January 2008.

[10] Council of Europe, "Draft Recommendation on the Protection of Individuals with regard to Automatic Processing of Personal Data in the Context of Profiling, The Consultative Committee of the Convention for the Protection of Individuals with regard to Automatic Processing of Personal Data". T-PD-BUR (2009) 02 rev 5 Fin, p. 5 (resulting from the 21st Bureau Meeting, Lisbon, 13-15 April 2010)

[11] N. J. King and P. W. Jessen, "Profiling the mobile customer–Privacy concerns when behavioural advertisers target mobile phones–Part I," *Computer Law and Security Review*, vol. 26, no. 5, pp. 455–478, 2010.

[12] D. C. Howe and H. Nissenbaum, "TrackMeNot: Resisting surveillance in Web search," *Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society*, pp. 417–436, 2009.

[13] H. Nissenbaum, "From Preemption to Circumvention: If Technology Regulates, Why Do We Need Regulation (and Vice Versa)," *Berkeley Tech LJ*, vol. 26, p. 1367, 2011.

[14] F. Brunton and H. Nissenbaum, "Vernacular resistance to data collection and analysis: A political theory of obfuscation," *First Monday*, vol. 16, no. 5, 2011.

[15] "Online Ad Spending Tops $100 Billion in 2012 - Forbes," *Forbes*. [Online]. Available: http://www.forbes.com/sites/roberthof/2013/01/09/online-ad-spending-tops-100-billion-in-2012/. [Accessed: 12-Mar-2013].

[16] "Do Not Track Is No Threat to Ad-Supported Businesses." [Online]. Available: http://cyberlaw.stanford.edu/node/6592. [Accessed: 12-Mar-2013].

[17] C. Castelluccia, M.A. Kaafar, and M.D. Tran, "Betrayed by Your Ads!" in *Privacy Enhancing Technologies*, Springer Berlin Heidelberg, pp. 1–17, 2012.

[18] J. Turow, J. King, C. Hoofnagle, A. Bleakley, and M. Hennessy, "Americans reject tailored advertising and three activities that enable it," *Available SSRN 1478214*, 2009.

[19] USA Today/Gallup Poll (2010). [Online] Available: http://gallup.com/poll/File/145334/Internet_Ads_Dec_21_2010.pdf. [Accessed: 14-Mar-2013].

[20] "Press Release: TRUSTe Announces 2011 Results From Behavioral Advertising Survey."[Online].Available:http://www.truste.com/about-TRUSTe/pressroom/news_truste_behavioral_advertising_survey_2011. [Accessed: 14-Mar-2013].

[21] C. Scott, "Our Digital Selves: Privacy Issues in Online Behavioral Advertising," *Appeal: Rev. Current L. & L. Reform*, vol. 17, p. 63, 2012.

[22] I. C. Government of Canada, "Questions and Answers." [Online]. Available: http://www.ic.gc.ca/eic/site/ecic-ceac.nsf/eng/gv00580.html. [Accessed: 15-Mar-2013].

[23] "Personal Information Protection and Electronic Documents Act," *Wikipedia, the free encyclopedia*. 29-Mar-2013.

[24] E. Felten, "FTC Perspective". Available: http://www.w3.org/2011/track-privacy/slides/Felten.pdf. 2011.

[25] R. Balebako, P. Leon, R. Shay, B. Ur, Y. Wang, and L. Cranor, "Measuring the effectiveness of privacy tools for limiting behavioral advertising," in *Web 2.0 Workshop on Security and Privacy*, 2012.

[26] "Commercial Privacy Bill of Rights.: Available: http://www.kerry.senate.gov/work/issues/?id=74638d00-002c-4f5e-9709-1cb51c6759e6&CFID=79733731&CFTOKEN=26547080. [Accessed 16-Mar-2013].

[27] "Well-Meaning 'Privacy Bill of Rights' Wouldn't Stop Online Tracking," *Electronic Frontier Foundation*. [Online]. Available: https://www.eff.org/deeplinks/2011/04/well-meaning-privacy-bill-rights-could-codify. [Accessed: 16-Mar-2013].

[28] J. Kesan, "Private Internet Governance," in *Loyola University Chicago Law Journal Symposium on Technology and Governance*, vol. 35, 2003.

[29] “DoubleClick nearing privacy settlements - CNET News,” *CNET*. [Online]. Available: http://news.cnet.com/DoubleClick-nearing-privacy-settlements/2100-1023_3-871654.html. [Accessed: 22-Aug-2013].

[30] “Federal Trade Commission,” *Wikipedia, the free encyclopedia*. 05-Apr-2013.

[31] “DoubleClick able to settle privacy suits - CNET News,” *CNET*. [Online]. Available: http://news.cnet.com/2100-1023-919895.html. [Accessed: 23-Aug-2013].

[32] “HTTP referer,” *Wikipedia, the free encyclopedia*. 16-Mar-2013.

[33] A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. J. Hoofnagle, “Flash cookies and privacy,” *SSRN Elibrary*, 2009.

[34] A. M. McDonald and L. F. Cranor, “A survey of the use of Adobe Flash local shared objects to respawn HTTP cookies,” *ISJLP*, vol. 7, pp. 639–721, 2012.

[35] M. Ayenson, D. Wambach, A. Soltani, N. Good, and C. Hoofnagle, “Flash cookies and privacy II: Now with HTML5 and ETag respawning,” *Available SSRN 1898390*, 2011.

[36] "What is the NAI's Policy on 'Flash Cookies' and Similar Technologies?" [Online]. Available: http://www.networkadvertising.org/nai-technology-policy. [Accessed: 8-March-2013].

[37] W. West and S. M. Pulimood, “Analysis of privacy and security in HTML5 Web storage,” *Journal of Computing Sciences in Colleges*, vol. 27, no. 3, pp. 80–87, 2012.

[38] “HTTP ETag,” *Wikipedia, the free encyclopedia*. 29-Mar-2013.

[39] P. Eckersley, “How unique is your Web browser?,” in *Privacy Enhancing Technologies*, Springer Berlin Heidelberg, pp. 1–18, 2010.

[40] “There is no such thing as anonymous online tracking.” [Online]. Available: http://cyberlaw.stanford.edu/node/6701. [Accessed: 06-Apr-2013].

[41] “Google Once Again Claims 67% Search Market Share,” *Search Engine Watch*. [Online]. Available: http://searchenginewatch.com/article/2244472/Google-Once-Again-Claims-67-Search-Market-Share. [Accessed: 16-Mar-2013].

[42] E. Adar, J. Teevan, and S. T. Dumais, “Resonance on the Web: Web dynamics and revisitation patterns,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1381–1390, 2009.

[43] R. Kumar and A. Tomkins, “A characterization of online browsing behavior,” in *Proceedings of the 19th international conference on World Wide Web*, pp. 561–570, 2010.

[44] “Datalogix privacy policy”. [Online]. Available: https://www.datalogix.com/privacy. [Accessed: 16-March-2013].

[45] D. D. Hirsch, “The Law and Policy of Online Privacy: Regulation, Self-Regulation or Co-Regulation?,” *Seattle University Law Review*, 2010.

[46] "Consumer Opt-out". Available: http://www.networkadvertising.org/choices/. [Accessed 22-March-2013].

[47] "Do Not Track". [Online]. Available: http://donottrack.us. [Accessed 22-March-2013].

[48] "Tracking Preference Expression (DNT)". [Online]. Available: http://www.w3.org/TR/tracking-dnt/. [Accessed: 22-March-2013].

[49] J. Mayer and A. Narayanan, "Re: Protecting Consumer Privacy in an Era of Rapid Change: A Proposed Framework for Business and Policymakers". [Online]. Available: donottrack.us/docs/FTC_Privacy_Comment_Stanford.pdf. [Accessed: 22-March-2013].

[50] L. Cranor, *Web privacy with P3P*. O'Reilly Media, Inc., 2002.

[51] S. E. Levy and C. Gutwin, "Improving understanding of website privacy policies with fine-grained policy anchors," in *Proceedings of the 14th international conference on World Wide Web*, pp. 480–488, 2005.

[52] J. Reagle and L. F. Cranor, "The platform for privacy preferences," *Commun. Acm*, vol. 42, no. 2, pp. 48–55, 1999.

[53] M. Olurin, C. Adams, and L. Logrippo, "Platform for privacy preferences (P3P): Current status and future directions," in *Privacy, Security and Trust (PST), 2012 Tenth Annual International Conference on*, pp. 217–220, 2012.

[54] O. Olurin, "Policy Merger System for P3P In a Cloud Aggregation Platform", *Masters thesis, University of Ottawa,* January 2013.

[55] G. Aggarwal, E. Bursztein, C. Jackson, and D. Boneh, "An Analysis of Private Browsing Modes in Modern Browsers," in *Proceedings of the 19th USENIX Security Symposium*, 2010.

[56] C. Dwyer, "Behavioral targeting: A case study of consumer tracking on levis. com," *Available SSRN 1508496*, 2009.

[57] "NoScript," *Wikipedia, the free encyclopedia*. 19-March-2013.

[58] P. Leon, B. Ur, R. Shay, Y. Wang, R. Balebako, and L. Cranor, "Why Johnny can't opt out: A usability evaluation of tools to limit online behavioral advertising," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pp. 589–598, 2012.

[59] "Chaff (countermeasure)," *Wikipedia, the free encyclopedia*. 05-Sept-2012.

[60] M. Fredrikson and B. Livshits, “RePriv: Re-imagining content personalization and in-browser privacy,” in *Security and Privacy (SP), 2011 IEEE Symposium on,* pp. 131-146. IEEE, 2011.

[61] J. Mayer, “Tracking the Trackers: Early Results.” [Online]. Available: http://cyberlaw.stanford.edu/node/6694. [Accessed: 21-March-2013].

[62] S. Peddinti and N. Saxena. "On the privacy of Web search based on query obfuscation: a case study of TrackMeNot." In *Privacy Enhancing Technologies, pp. 19-37. Springer Berlin Heidelberg,* 2010.

[63] “Social Media Targeting,” *Wikipedia, the free encyclopedia*. 13-Mar-2013.

[64] E. Adar, J. Teevan and S.T. Dumais, "Large scale analysis of Web revisitation patterns," In *26th Annual Conference on Human Factors in Computing Systems,* 2008.

Appendices

A. Google Ads Preferences Root Categories

Arts & Entertainment
Autos & Vehicles
Beauty and Fitness
Books & Literature
Business & Industrial
Computers and Electronics
Finance
Food & Drink
Games
Hobbies & Leisure
Home & Garden
Internet & Telecom
Jobs & Education
Law & Government
News
Online Communities
People & Society
Pets & Animals
Real Estate
Reference
Science
Shopping
Sports
Travel
World Localities

B. ProfileMeNot Instructions

B.1 First batch of instructions

Bird's eye view of the experiment and its importance
This section is not required for the experiment, it is just for your information.

As you are browsing the web, advertising networks and others (such as Google Adwords, Yahoo-bing, etc.) track you on different sites through different tracking technologies. The most commonly used and simplest tracking technology is the cookie. A cookie is a small piece of data sent from a website and stored on your browser.

As you are tracked, a behavioral profile is built for you based on your browsing behavior. This behavioral profiling is a privacy concern. For example, it might be inferred that you have an unhealthy life style. That information about your unhealthy lifestyle might be sold to an insurance company, and when you apply to get the insurance, you will be denied (this is just an example, not necessarily happening).

There are different solutions to solving the privacy issue. Most solutions (AdBlock, NoScript etc.) do not necessarily block tracking. In the next phase of the experiment, I will be sending you my solution with an explanation of how my solution works. My solution is implemented as a FireFox add-on. To know what an add-on is, please refer to the FAQ section.

In brief, the add-on will be 'simulating' a users behavior by visiting websites on behalf of the user. This is in order to create noise in the profile that is built on you. I have tried my best to block controversial material such as Pornographic sites and sensitive religious websites. The add-on will not trace you and will not profile you. It is built to obfuscate the profile that is being built on you. Also to note that my add-on was built based on an add-on spearheaded by NYU called TrackMeNot(TMN). TMN was created by Daniel C. Howe, Helen Nissenbaum and developed by Vincent Toubiana and is released under a creative commons license.

Instructions on the first step of the experiment

1) On your current browser, please visit http://www.google.com/ads/preferences
Once you are on that page, please copy and paste the 'Your Categories' and 'Your demographics' section. Don't worry if there is nothing there, please still send it. If you have trouble with copy/pasting or going to the link, contact me. Please name your file as : NAME, LAST NAME, DATE, GOOGLE ADS PREFENCE'

2) Once again, on your current browser, please visit http://info.yahoo.com/privacy/us/yahoo/opt_out/targeting/ and copy and paste 'Interest categories' , 'Your activities' and 'Your computer and cookies' into a document and

email it to me. Once again, if there is nothing there, please still email it to me. NAME, LAST NAME, DATE, YAHOO ADS PREFENCE'

3) If you have Firefox installed, please see 3*. If not, please go to http://www.mozilla.org/en-US/firefox/new/ and install Firefox. Please use Firefox for the next 5-6 days. On the 5th -6th day, I will be sending you my solution with instructions on how to install it.

3*) If you have Firefox already installed and Firefox is your primary browser, then please create a new profile. See the appendix on how to create a Firefox profile.

4) On your Firefox, please visit www.google.com/ads/preferences, and under '**What to do'**, click on '**enable now**'. *DO NOT click on 'opt-out'.*

5) On your Firefox, please visit http://info.yahoo.com/privacy/us/yahoo/opt_out/targeting/ and if you see a yellow button that says 'OPT IN', click on it, if it says 'OPT OUT', DO NOT CLICK ON IT! ☺

6) After five-six days of browsing, (a reminder will be sent), please redo steps one and two!

7) Once you have done all the above steps, send me a confirmation email.

8) Great! We are done for this first step of the experiment. You may now use your Firefox as your primary browser for five days. After that, I will send you the Firefox add-on with instructions on how to install it.

Frequently Asked Questions (FAQ)

1) **What is a Firefox add-on?**
   Add-ons are installable enhancements to the Mozilla Foundation's projects, and projects based on them. Add-ons allow the user to add or augment application features, use themes to their liking, and handle new types of content. (Wikipedia).
2) **Will tracking companies be tracking me with this add-on?**
   No, many companies are already tracking you; my add-on will help you against companies that build a profile on you. The add-on will not trace you in any form.
3) **Can I delete the add-on when the experiment is done?**
   Yes, you can. Instructions will be given when the add-on is sent.

To make a new Firefox profile, please follow one of the following, based on your current Operating system.

For **windows** users:
Close all running Firefox versions (File → Exit)
Go to windows Start Menu (bottom left button) and then select 'Run'.
Type in firefox.exe –P
Click on the 'Create Profile' button and follow the instructions.
Once the profile has been created, click on 'Don't ask at startup' and open the new profile. If you wish to still use your old profile for whichever reason, do not click on 'Don't ask at startup'.
Once we are done with the experiment, follow steps 1 through 3, select your old profile and delete the older profile. For more detailed information, see :http://kb.mozillazine.org/Creating_a_new_Firefox_profile_on_Windows

For **Mac** users:
1) On the menu bar, click on the Firefox menu and select Quit Firefox

2) Navigate to */Applications/Utilities*. Open the **Terminal** application.

3) In the Terminal application, enter the following: /Applications/Firefox.app/Contents/MacOS/firefox-bin –p

4) Press Return (enter)

5) To start the Create Profile Wizard, click Create Profile... in the Profile Manager.

6) Click Next and enter the name of the profile. Use a profile name that is descriptive, such as your personal name. This name is not exposed on the Internet.
7) You can also choose where to store the profile on your computer. To choose its storage location, click Choose Folder.... **Warning:** If you choose your own folder location for the profile, select a new or empty folder. If you choose a folder that isn't empty and you later remove the profile and choose the "Delete Files" option, everything inside that folder will be deleted.

8) To create the new profile, click Done.

9) For the duration of the experiment, please use this profile.

For more information see: http://support.mozilla.org/en-US/kb/profile-manager-create-and-remove-firefox-profiles

B.2 Second batch of instructions

PROFILEMENOT
Killing two stones with one bird
(Based on TrackMeNot, created by Daniel Howe and Helen Nissenbaum, developed by Vincent Toubiana)
Second stage of the experiment
Created and Developed by: Hadi Sajjadpour
Under the supervisions of: Evangelos Kranakis and Carlisle Adams

⚠***Important: At times, when restarting your computer or when going from sleep back to on, ProfileMeNot might not function. Simply right click on it on the bottom left of your Firefox browser where it says 'PMN', click on disable, then re-enable it. To hide a tab, see 'How to hide tabs at the end of the document'***

1) First, in a text document, indicate, when you installed Firefox, how many days you have been using it and how often (mostly using this browser etc.) and name it : NAME, FIREFOX USAGE, DATE. Please email this document at hadi.sajjadpour@gmail.com
2) If you have been using Firefox as your main browser in the past few days, please go to http://www.google.com/ads/preferences, copy and paste the entire page into a document, and email me the document. Name the document as the following: NAME, LAST NAME, DATE, GOOGLE ADS PREFERENCES. If you have not been using Firefox, please call me at 613-240-8095 so I can give you additional instructions. Email at hadi.sajjadpour@gmail.com
3) Please do the same as step 2, but for Yahoo. Please go to http://info.yahoo.com/privacy/us/yahoo/opt_out/targeting/ , copy and paste the entire document and email it to me. If you have a yahoo account, please log into your Yahoo account before emailing it to me. Please name this file as the following : NAME, LAST NAME, DATE, YAHOO ADS PREFERENCES
4) Download the add-on that has been sent to you in the email.
5) Open your Firefox browser.
6) If you have Adblock plus, Adblock, TrackMeNot, NoScript, Adjail or any other add on that will disable cookies. Please disable them for the duration of this experiment. See video on how to disable them or call me for more instructions.

7) Drag and drop the add-on 'ProfileMeNot.xpi' into Firefox.
8) Firefox will ask you to restart, restart Firefox.
9) Restarting Firefox now, on your bottom left, you will see something like 'PMN(0)'.
10) It is most likely already enabled, however, if it says 'Off', right click, and click on 'Enabled'.
11)
11) Set the frequency to 1 link per minute.
12) Click on OK.
13) ***At times, when restarting your computer or when going from sleep back to on, ProfileMeNot might not function. Simply right click on it on the bottom left of Firefox, click on disable, then re-enable it.***

14) Congratulations! You now have ProfileMeNot installed on your Firefox browser. Please read step 15 for what needs to be done after 3 and 6 days of using ProfileMeNot. To see what happens, see below under, 'What does ProfileMeNot do?'.

15) Please use ProfileMeNot for around 6 or 7 days. After the third day and after the final 6th or 7th day, please send me the following:

a) Please go to http://www.google.com/ads/preferences, copy and paste the entire page in the document and email it to me. Name it as: NAME, LAST NAME, DATE, GOOGLE ADS PREFERENCES, USING PMN X(Where X is either 1 for the first 3 days, or 2 for the second three days)

b) Please go to http://info.yahoo.com/privacy/us/yahoo/opt_out/targeting/ copy and paste the entire page into a document and email it to. Name it as: NAME, LAST NAME, DATE, YAHOO ADS PREFERENCES, USING PMN X

c) On your browser, on the bottom left, right click on PMN and click on options. Click on show log. Copy and paste the entire log and email me the log. The log contains only information that the addon has done, nothing from your browsing.

What does ProfileMeNot do?

You will not see another tab appearing as you browse the web. ProfileMeNot will browse the web alongside with you. The goal of ProfileMeNot is to create noise in the behavioral profile that is built on you. The reason I ask for the Google Ads Preferences before and after using PMN, and also in the middle, is to measure how much noise has been introduced. Different volunteers will be getting different versions of ProfileMeNot with different values to input.

The way ProfileMeNot gets the links is by doing a web search using the Bing engine. It then takes the returned links, parses them and based on some algorithms, will visit one of them in the tab that appears. The queries are taken from RSS feeds. You can

change the RSS feeds under the RSS feed textbox in ProfileMeNot options, however, for the purposes of this experiment, please do not change the feeds.

We have tried our best to avoid any controversial material to be visited during your browsing session. To add to the list of words that you don't want the search engine to query, add them to the comma separated '**List of black-listed keywords'.** Separate each word by a comma. There have been some links added for your convenience.

We have also tried our best to avoid any tube or video related sites to avoid consuming bandwidth and creating noise during your browsing session. However, at times, if it does happen, we appreciate your patience.

How to hide a tab

As the Addon will be adding a tab and browsing at the same time as you, you might want to hide the tab so you don't see the changes. To do this:

1) Please visit https://addons.mozilla.org/en-us/firefox/addon/hidetab/ and install the HideTab add-on.
2) Restart Firefox
3) The hide a tab, then simply right click on tab, and select Hide Tab.
4) To make tabs reappear, right click on the tabs pane, and select restore all tabs or specify which tab you want re-opened.