Privacy Awareness about Information Leakage: Who knows what about me?

Delfina Malandrino
delmal@dia.unisa.it

Andrea Petta
andrpet@gmail.com

Vittorio Scarano
vitsca@dia.unisa.it

Luigi Serra
luigser@gmail.com

Raffaele Spinelli
spinelli@dia.unisa.it

Balachander Krishnamurthy
bala@research.att.com

ABSTRACT

The task of protecting users' privacy is made more difficult by their attitudes towards information disclosure without full awareness and the economics of the tracking and advertising industry. Even after numerous press reports and widespread disclosure of leakages on the Web and on popular Online Social Networks, many users appear not be fully aware of the fact that their information may be collected, aggregated and linked with ambient information for a variety of purposes. Past attempts at alleviating this problem have addressed individual aspects of the user's data collection. In this paper we move towards a comprehensive and efficient client-side tool that maximizes users' awareness of the extent of their information leakage. We show that such a customizable tool can help users to make informed decisions on controlling their privacy footprint.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Protocols—*applications*

General Terms

Measurement, Performance

Keywords

Privacy-awareness/leakage; enhancing technologies

1. INTRODUCTION

Given the increasingly important role of online communication in people's everyday life, enhancing users' privacy protection is a critical issue. Increasing amounts of both personally identifiable information (PII) and sensitive (e.g., medical, financial and family) information continue to be leaked [7, 9, 14]. The situation has been exacerbated through the introduction of free popular services, such as on Online Social Networks (OSN), and the ability of advertising companies to deliver targeted advertising. Privacy can be undermined by third parties [2]. Users effectively pay for these free services through micro payments of ever-greater amounts of personal information.

The online marketing methods of network advertisers have given rise to concerns about user's privacy [1]. Although the practice of tracking individuals' online activities increases the effectiveness of the marketers' campaigns, it also undermines the privacy of users, mainly because it relies heavily on users' personal information. Pseudo-anonymous data collected and linked with PII such as email addresses and credit card number, may be sold by aggregators. The possessors of such data may use it for identity theft, social engineering attacks, online and physical stalking and so on[1] [5, 16].

This paper makes several contributions. First, we show how NoTrace [11, 12], a privacy-enhancing tool: (1) Fully addresses the most important requirements that tools have to exhibit to protect privacy on the Web, that is comprehensiveness, support and awareness, performance and effectiveness [17] (2) Displays in real time, that is during a browsing session, leakages of personal information (3) Raises awareness of measures to safeguard personal data and search habits (4) Improves privacy of Web users. Second, we show that NoTrace can detect *more* information leakage than other popular privacy tools at a *lower cost*. Third, we design a hierarchy of the most important privacy threats analyzing the ways in which personal and sensitive information are sent to third party sites. We derive an ordering of the importance of the tools according to the countermeasures they provide and their effectiveness in limiting the disclosures of important information. Fourth, we show that, by linking pieces or bits of personal information leaked towards different third party sites, it is possible to identify users and derive their interests and browsing habits. We show how NoTrace is able to give real time information about which aggregators have what portion of users' personal data.

2. PRIVACY AWARENESS AND NOTRACE

We summarize privacy awareness as encompassing the perception of: (1) *Who* is tracking and collecting personal information (2) *When* information is collected (3) *What* information other entities receive, store and use (4) *How* pieces

[1] `http://www.priv.gc.ca/media/nr-c/2012/nr-c_120925_e.asp`

WPES'13, November 4, 2013, Berlin, Germany.

http://dx.doi.org/10.1145/2517840.2517868.

of information are processed to potentially build detailed users' profiles.

Although the complexity and the efficacy of data mining technologies are growing quickly to increase the effectiveness of behavioral advertising, the awareness of privacy erosion is growing slowly [7]. In this paper we show how NoTrace informs users about which pieces of personal information are disclosed to third party entities. As more users learn about their information leakage they may be able to make better decisions about controlling their privacy [13].

NoTrace, a Firefox add-on included in the Privacy & Security Category of the Mozilla Community[2], relies on a modular architecture. This modularity represents the key factor to provide measures for privacy protection for many privacy threats, by also guaranteeing efficiency and effectiveness. From the technical point of view, NoTrace leverages the Cross Platform Component Object Model (XPCOM) framework[3], that allows the development of modular software and provides tools to create, assemble and manipulate components at run-time. The implemented components manages HTTP requests/responses headers, on-the-fly transformations before the browser rendering, and the traditional URL-based blocking mechanism.

2.1 NoTrace Requirements

Support and comprehensiveness. NoTrace supports users' needs through several privacy settings that can be fine-tuned according to experience and expertise [12]. It is able to address several privacy threats by providing opportune countermeasures, whereas each of them is singularly provided by other popular add-ons in this field, that sometimes are in conflict with each other. It also does not involve performance slowdown of the browser, that may occur if multiple tools have to be installed to provide the same countermeasures. An experiment verifying this claim is discussed in Section 3.1.

We extended the set of provided measures described in our earlier works [11, 12] with techniques to block requests for large advertising companies, or to alter the browser fingerprint information [3] for requests to third party sites. Additionally, we implemented a new *"External-filtering"* mechanism to access to the stream of bytes received by the browser immediately before the rendering of the Web page. It allowed to implement new protection measures, such as those that look at Cookies and Referer fields set in external JS codes. No other tool has harnessed this type of filtering before, leveraging URL-based filtering mechanisms only.

Awareness and full control. To educate users about what private information they leak towards third party aggregators and the information that is inferred based upon their behavior, we deployed in NoTrace specific awareness modules. Specifically, NoTrace shows which information are leaked towards third party entities, for each visited Web site and which fraction of users' personal data is known and shared by many popular third party entities. No other tool has provided this type of awareness before.

Performance and effectiveness. Excessive delays experienced by users when using a tool may involve its abandonment just after first use [4]. We tested the effectiveness and the impact on the user's experience of this improved version of NoTrace, because of some changes we made in this work to speed up performance. The positive results are presented in Section 3.1.

[2] `https://addons.mozilla.org/en-us/firefox/addon/notrace/`

[3] `https://developer.mozilla.org/en/XPCOM`

3. A COMPARATIVE STUDY

We compare NoTrace and other popular tools that are comparable in terms of functionalities: AdBlock Plus (`http://adblockplus.org`), NoScript (`http://www.noscript.net`), Ghostery (`http://www.ghostery.com`), and RequestPolicy (`https://www.requestpolicy.com`).

All tools provide functionalities to filter ads and to block third party requests. Techniques for HTTP removal, 3d-party and Opt-out cookie blocking, HTML5 Local Storage managing and Web bug filtering are fully supported by NoTrace, and only partially by the other tools. All tools implement the URL-based blocking mechanism, whereas only NoTrace provides mechanisms to inspect in real time the content of Web pages. Finally, awareness about data leakage is provided only by NoTrace, while crowdsourcing of filtering rules by NoTrace and AdBlock Plus only.

Our comparative study will cover both the impact on users' perceived experience and performance (Section 3.1) and the effectiveness of the tested tools in terms of false positives and false negatives due to the filtering rules (Section 3.2). We show that NoTrace provides privacy protection at a lower cost and without degrading page quality or cause functional breaks.

3.1 Impact on User Experience

Following [6], our data set consists of the top-100 Web sites from 15 Alexa categories (`http://www.alexa.com`). Augmenting the Firefox browser by the Pagestats extension (`http://www.cs.wpi.edu/~cew/pagestats`) we retrieved 1500 pages that involved over 200,000 URLs to be analyzed.

In NoTrace we enabled techniques filtering out ads, Web bugs, hidden 3d-party scripts, requests for 3d-party domains and aggregators and we compared each tool's behavior individually to the baseline experiment without any tool installed ('NoAddons'). We used different browser profiles and performed experiments sequentially.

3.1.1 Response time results

We compared how the tested tools perform in terms of mean response times when applying the filtering capabilities on our data set. We calculated the gain in terms of response time when third party objects are being removed from users' requests. We computed the objects retrieved on a page when filtering is applied, against objects retrieved under normal conditions (i.e., the "NoAddons" experiment).

NoTrace shows better behaviors than those exhibited by AdBlock Plus and Ghostery, but it has a greater response time when compared with NoScript and RequestPolicy (overhead of almost 600ms for both). Specifically, NoTrace is able to save (on average) about 1.9 seconds against the baseline (3832ms vs. 1940ms). Additionally, it is able to block unwanted objects and save 35% of the total MegaByte transferred in downloading Web pages. The saved bytes for NoScript, RequestPolicy, AdBlock Plus and Ghostery are 54%, 57%, 23%, 29%, respectively.

The principal reason why NoScript and RequestPolicy are faster is the large number of resources blocked via their filtering rules. NoScript blocks, regardless of the real danger of detected objects, *all JavaScript code*, even those that are essential to the correct behavior of the page, while Request-

Policy has a stricter set of rules, avoiding the page break for very popular Web pages only because they are included by default in the startup whitelist. We show empirically in Section 3.2 that NoScript and RequestPolicy strict policies negatively impact the quality and the functionality of the Web pages returned, drastically compromising the user's Web experience.

3.1.2 Browser performance results

Among the studied privacy protection tools, none is able to fully address many privacy threats. Therefore, multiple add-ons have to be installed, involving possible performance degradation of the browser[4]. To study this we compared the performance of Firefox when loading up to 8 add-ons (i.e., AdBlock, NoScript, Ghostery, RequestPolicy, Taco, RefControl, PrivacyChoice and TrackMeNot) with specific techniques (i.e., ads and Web bugs filtering, 3d-party JS code execution blocking, opting-out from *ad*-networks, HTTP Referer blocking) against its performance when only NoTrace is loaded as a way to provide "all in one" functionality. Results showed an higher Firefox loading time for the multiple-installations (1260ms vs 360ms for NoTrace).

We tested the memory footprint during a reasonable facsimile of several hours of Web browsing. We ran the MemBench script[5], which is a memory test benchmark that opens 150 popular Web sites, one per tab. After closing 150 tabs, Firefox resident memory consumption (measured through the "about:memory" monitoring tab) with multiple extensions is 2.8x larger than Firefox with only NoTrace installed. After closing the tabs, the initial allocated memory was not fully released doubling, conversely, its initial value.

We analyzed the memory consumption separately experienced by each tool. AdBlock Plus starts with the highest memory since it needs to load the subscription list. Ghostery shows worse results since the resident memory at the end of the experiment was 4 times higher than the startup value. NoScript and RequestPolicy show better memory consumption values due to the high number of blocked resources.

We also tested how *multiple installations* of tools may involve a larger consumption of the memory. The final allocated memory for the *NoTrace single installation* was 120MB. Installing NoTrace with AdBlock, and Ghostery, involved an increase of the value of the not released memory up to 300 MB. We not tested RequestPolicy and NoScript here, since the both the resident memory and the final allocated memory value drastically decreased, due to the number of filtered resources and not because of better performance. Overall, by using NoTrace alone we can save on average 60% of the memory; an amount that becomes more significant in the mobile environment.

3.2 Effectiveness

We manually analyze the 1400 embedded resources in the top 10 Alexa News category sites. We whitelisted all CDN domains to distinguish objects needed for proper functioning; we call this technique *intelligent filtering*.

3.2.1 Results

We analyzed False Positives (FP) and False Negatives (FN) for all tested tools. Due to space limitations we will discuss only NoTrace errors in detail. Table 1 shows in column 7, the number of FP detected when applying *intelligent filtering* (i.e., IF in Table 1) and without considering domains that serve their content for first party sites (i.e., NoIF). As an example, for the `foxnews.com` Web site, its content also comes from a third party entity, that is `fncstatic.com`, mostly serving Web images. Thus, by indiscriminately blocking all third party resources, the quality of the page could be degraded without any privacy improvement, leading to a large number of FP.

With *intelligent filtering* we will avoid all FP. The same argument applies to all the analyzed Web sites. NoTrace's FN, instead, can be due to: (i) First party requests for resources that are not available in the DOM (ii) Objects served by CDNs of first party sites, and (iii) 3d-party requests for resources that are not available in the DOM.

The first category includes requests for Web bugs (i.e., `us.bc.yahoo.com/b`). NoTrace is not able to block them, as its technique to filter Web bugs looks at the height and weight properties of the images available in the DOM of the requested Web page. Similar to AdBlock Plus, we allow users to add an ad-hoc filtering rule to block them.

The second category includes errors due to the inclusion of the CDN servers into the whitelist because of their role in serving needed content for the requested Web pages[6].

The third category includes errors due to third requests for resources not available in the DOM. Here, the high number of errors is due to a request for a JS code that loads a certain number of both harmless and malicious scripts (13 out of 16 errors are Web bugs for the `weather.com` Web site). If we remove the loader we can avoid tracking, but also break the quality of the Web page, since the harmless scripts are used for page formatting and additional site's functionalities. A feasible solution requires examine the URL to extract the internal scripts, block the unwanted ones, and then resubmit the modified URL.

In summary, as shown in Table 1, the incidence of FP and FN for NoTrace is low, while as expected, NoScript and RequestPolicy exhibit the highest number of errors. To compare tools, we also plotted the number of FP and FN of the analyzed sites. Fig. 1 shows NoTrace's better behavior and the worst behavior of both NoScript and RequesPolicy with an extremely high FP. In two cases RequestPolicy has over 100 errors, and few FP exist only when the corresponding domains are in the whitelist. Properly configuring the whitelist requires more expertise than an average user can be expected to have.

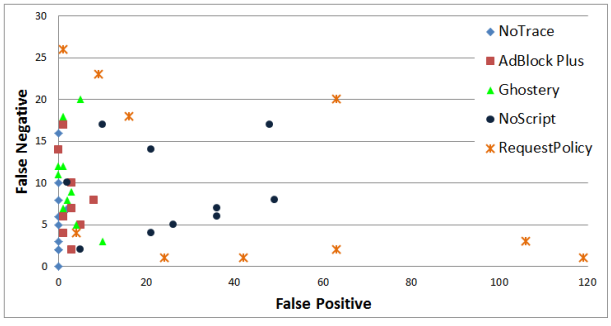

Figure 1: Analysis of FP and FN after blocking.

[4] `http://blog.mozilla.org/addons/2010/06/14/improve-extension-startup-performance/`

[5] `http://gregor-wagner.com/tmp/mem`

[6] `http://i.cdn.turner.com/cnn/.e/img/3.0/1px.gif`

Table 1: Effectiveness on popular Web sites: FP and FN. For NoTrace we also consider whitelisted CDNs.

Web Site	Web site's CDNs	AdBlock Plus		Ghostery		NoTrace		NoScript		RequestPolicy	
		FP	FN	FP	FN	FP (IF/NoIF)	FN	FP	FN	FP	FN
news.yahoo.com	yimg.com	1	10	10	3	0/10	0	10	17	9	23
edition.cnn.com	turner.com	0	34	1	12	0/3	2	21	14	1	26
weather.com	imwx.com	3	7	5	20	0/29	16	36	7	16	18
reddit.com	redditmedia.com redditstatic.com	3	3	2	8	0/2	3	5	2	24	1
my.yahoo.com	yimg.com yahoapis.com	3	5	3	9	0/2	7	2	10	4	4
bbc.co.uk/news	bbcimg.co.uk bbci.co.uk	1	8	0	11	0/14	5	36	6	106	3
foxnews.com	fncstatic.com	8	6	1	18	0/35	2	49	8	63	2
nytimes.com	nyt.com	1	11	0	12	0/7	10	48	17	63	20
huffingtonpost.com	huffpost.com	1	23	1	7	0/4	6	21	4	42	1
guardian.co.uk	guim.co.uk	5	10	4	5	0/3	8	26	5	119	1
Total		26	117	27	105	1/109	59	254	90	447	198
Recall/Precision		0.93/0.77		0.89/0.74		1.00/0.86		0.66/0.79		0.60/0.81	

4. INFORMATION LEAKAGE STUDY

We now explore the manners of leakage through which personal and sensitive information are sent to *third party sites*, such as third party Cookies, Referer header, Web bug, third party JavaScript, Redirect Tracking, or advertisements. The "*Redirect Tracking*" leakage vector uses the HTTP redirect mechanism to redirect a user to the URL of a third party site. We classify the countermeasures for the most popular threats we found. NoTrace is able to detect the key leakages at a lower cost.

4.1 Methodology

To analyze the leakage of personal and sensitive information we used 18 (sub)categories of Alexa and selected the top-10 sites that allow users to register. The categories are: Health, Travel, Employment, OSN, Arts, Relationships, News, PhotoShare, Sports, Shopping, Games, Computer, Home, Kids_and_teens, Recreation, Reference, Science, and Society. We extended the data set used in Section 3.1 to consider two categories–OSN and Relationships–with a large number of registered users, one–Employment–that involves users supplying private information, and one–PhotoShare–that may involve leaks due to potentially harmful specific actions, such as inputting content. We set up accounts with the corresponding first party sites rather than signing in via a third party account. We also enabled the option "Remember me" for sites that allowed that option, to study if private information are stored and then sent to third party sites.

We added detailed information to the 180 accounts we built, including full name, email address (required for all accounts), Date Of Birth (DOB), Social Security Number (SSN), zip code, home address, personal cellphone, school and general education information, sexual orientation, political and intellectual beliefs, general interests (music, movies, and travel). They represent the bits of private information that may be leaked towards 3rd-party sites.

We then created a log of typical interactions between the user and the sites. We included actions that may uniquely identify the users from (a) search terms[7], (b) browser habits, (c) preferences about music, movie and books[8], and (d) the structure of their social networks [15]. We used the following six types of online users' interactions:

1. Account Login and Navigation. We logged in on all 180 sites and analyzed information leakage due to 3d-party cookies. We also visited 4 or 5 embedded links per page, to reflect typical navigation of a user [8].

2. Viewing/Editing Profile. To reflect the most common actions performed by users on OSN we analyzed the following actions: viewing one's own profile and editing it, viewing 5 friend's profiles, writing on the "Timeline" of 2 of them.

3. Searching the Web for Sensitive Terms. We searched using `google.com` for 20 terms in 7 sensitive categories: Health (3), Travel (5), Jobs (2), Race and Ethnicity (2), Religious beliefs (3), Philosophical and Political beliefs (4), Sexual orientation (1). For each search term we also navigated through the first 2 search result pages.

4. Popular search. We chose 10 keywords from the top Google searches in 2012[9] and Google Trend Web pages[10].

5. Inputting and Like-ing content. For Inputting content we analyzed the following actions: post and reply to questions on forums (2 actions), reply to dating messages (1 action), upload pictures (1 action). For Like-ing content we analyzed the following actions: "Like" on Facebook (2 actions), "Share" via Facebook (2 actions), "+1" on Google Plus (2 actions), "Share" via Google Plus (2 actions).

We used Selenium (`http://seleniumhq.org/`) to automate tests, logging HTTP headers and saving both the HTML pages and JS codes. We generated a set of strings related to the personal information we added to the 180 accounts at their creation time, and the sensitive terms that we searched for. We searched the Selenium logs for these strings and removed false positives by hand. When leakage occurred, we recorded the leaked information, the manner of leakage, and the third party destinations.

4.2 Information leakage results

4.2.1 Categorization of the most important leakages.

By extending the work done in [7], we identified the following leaked bits (newly identified leakages are in bold): Full name, Email, ***IP address***, Country, Region, City, Zip code, ***Education*** and Employment, Gender, Age, ***DOB***, ***Interests*** (Movie and Music), ***Sexual orientation***, Political and religious beliefs and ***browser fingerprint*** information. Using categorization in [7], we organize these bits into High, Medium, and Low categories by taking into account their degrees of sensitivity and identifiability.

We observed both first party sites leaking the bits and 3d-party sites that receive the leaked bits. A total of 44 first

[7] `http://www.nytimes.com/2006/08/09/technology/09aol.htm`
[8] `http://www.cs.utexas.edu/~shmat/shmat_oak08netflix.pdf`
[9] `http://www.google.com/zeitgeist/2012/#the-world`
[10] `http://www.google.com/trends/hottrends`

party sites out of 214 leaked private information[11]. Specifically, for the *High* Category, health terms are leaked in 3 of the 4 sites studied. For the *Medium* Category we derived that an important bit leaked by a number of sites was the user's full name. This leakage raises concerns when this bit is combined with sensitive terms. In Job and Travel searches, 4 out of 5, and 6 out of 7 of the studied sites show leakage respectively (actions discussed in Section 4.1). Health information could be combined with user's personal information and create difficulties while seeking health insurance. Job information combined with user's personal information can lead to privacy attacks such as identity theft[12].

Our analysis showed that the most important vehicles through which all types of categories' bits are leaked are the Referer HTTP field and Web bug; blocking them would yield better privacy protection.

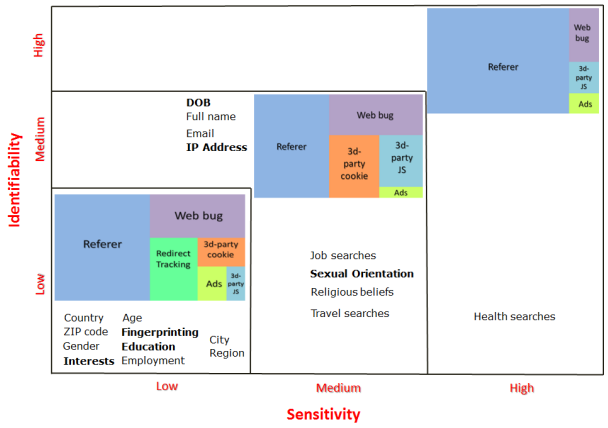

Figure 2: Distribution of the most important leakage vehicles across the Low/Medium/High categories.

In Fig. 2 we show the distribution of the most important leakage vehicles across the Low, Medium, and High categories. We highlight the new leaked bits discovered in our study in bold as compared to [7]. For all categories the Referer is the most used vehicle to track users. Only for the Low category we saw differences across all 6 manner of leakages.

4.2.2 *Classification of the tools to improve privacy.*

Results aforementioned described are obtained by navigating without any privacy protection tool. We repeated the same automated interactions from Section 4.1 when privacy tools are used. We found Ghostery still leaks full name, city, zip code, region, gender, age, DOB, IP Address and browser fingerprint. NoScript and RequestPolicy had less leakage, since their stricter filtering rules. Specifically, NoScript leaked zip code, gender, age, IP Address, while RequestPolicy full name, region and DOB. Overall, in this analysis we found out that there was no leakage in Header or URL for NoTrace, whereas AdBlock Plus had a total of 82 Header leakages (that include leakages via Cookie and Referer), with most in job/religious/political and 10 URL leakages (that include leakages via Web bug, 3d-party JS, Ads, Redirect Tracking), mostly political and job. Ghostery had a total of 102 Header leakages, with most in health/job/political, and 49 URL leakages, mostly travel and political. NoScript had a total of 33 Header leakages (with most in health searches) and 6 URL leakages, and finally, RequestPolicy had a total of 11 Header leakages and 3 URL leakages. NoTrace is most effective in reducing the diffusion of both personal information and sensitive search terms, as the zero values for both Header and URL leakages.

[11]Details are available in our extended technical report [10]

[12]`http://www.job-hunt.org/privacy.shtml`

5. WHO KNOWS WHAT

We now see if it is possible to build a detailed profile about users by collecting and linking private information bits that users disclose online from diverse sources. To analyze what fraction of a user's profile is known by the top-10 aggregators, we instrumented Selenium to perform specific actions: (1) Logging into all 180 accounts (2) Viewing and editing all 10 profiles from the OSN category, post comments and messages, share documents with "friends" (3) Search on all 10 shopping sites from the Shopping category, add items to shopping carts (without payment), create lists, "Like" content (4) Search on all 10 Job-related sites from the Employment category, sign up for email alerts (5) Search on all 10 Health sites from the Health category, post comments (6) Search on all 10 Travel sites from the Travel category, book travel arrangements (without payment), visit Google maps site for itineraries, share with friends (via email and OSNs) (7) Reply to messages on 5 out of 10 Web sites of the Relationships category, that not required a Premium account (8) Create Photo Galleries on the `photobucket.com` Web site, upload images, add comments, share with friends, "Like" content (9) Watch videos on the `youtube.com` Web site, post comments, share with friends, "Like" content (10) Play songs on the `last.fm` Web site, post comments, share with friends, "Like" content. All interactions were logged by Selenium. We then examined its logs for users' private information leakage and fractions of this information known by the top-10 aggregator servers.

5.1 Results

We used the top-10 leak recipients identified in our data set (Table 2). To analyze results we used the same method of Section 4.1. We extended the set of strings to also look at sensitive Health terms (i.e., Pregnancy, Depression, Breast Cancer), Job terms (i.e., Analyst, Senior Analyst in New York), Travel terms (i.e., traveling from Napoli Capodichino to New York (JFK) and travel dates), music, book, and movie interests (i.e., Black Eyed Peas, Internet Traffic Measurement, and Viva l'Italia movie).

In Table 2 we report which bits are received by each aggregator. The fraction of known bits, i.e. number of received bits respect to the total number of analyzed bits, ranges between 12% for pubmatic.com, to 87% known by google-analytics.com. Surprisingly, the Health terms are leaked to almost all top-10 aggregators. Google Analytics is the top recipient of the leakages, since it receives 87% of leaked bits.

Linking of several exchanges, *ad*-servers, or *ad*-networks (i.e., daisy chaining[13]) can increase chances of building detailed dossiers about users. We found in our study many communications among aggregators with leakage of private information. Column 1 of Table 3 shows the count and the

[13]http://www.masternewmedia.org/online-advertising-management-ad-network-defaulting-and-daisy-chaining-for-ad-revenue-optimization/

Table 2: Building a profile from pieces of private and sensitive information.

Aggregator	Email	IP Address	Country/Region/ City	Zip Code	Gender	Age	DOB	Interests	Health/ Job	Religious/ Political	Sex Orient.	Travel	Known bits [%]
doubleclick.net	–	√	√/√/√	√	√	√	√	√	√/√	√/–	–	√	81
google-analytics.com	√	–	√/√/√	√	√	–	√	√	√/√	√/√	√	√	87
scorecardresearch.com	√	–	√/–/√	√	√	√	–	√	√/√	√/–	–	√	69
adnx.com	–	–	–/√/√	√	√	√	–	–	–/–	√/–	–	–	37
yieldmanager.com	–	–	–/–/√	√	√	√	–	–	√/√	√/–	–	–	44
2o7.net	–	√	√/–/√	–	√	–	–	–	√/√	–/–	–	–	37
crwdcntrl.net	–	–	–/–/√	–	√	√	–	–	√/–	–/–	–	–	25
pubmatic.com	–	√	–/–/–	–	–	–	–	–	√/–	–/–	–	–	12
2mdn.net	–	√	√/√ /√	√	√	√	–	–	√/–	–/–	–	√	56
imrworldwide.com	–	–	√/√/√	–	√	–	–	–	√/–	√/–	–	–	37

first party sites contacted. Column 2 shows the first and second aggregator involved in daisy chaining, while the last column lists the bits leaked. We identify daisy chaining by examine HTML body which includes an IFRAME triggering an auto-request to the first aggregator. Further aggregator requests linkage can be tracked via the Referer header.

Table 3: Data leakage through daisy chaining.

Count/	Aggregators		Bits leaked
1st party sites	1st Aggr.	2nd Aggr.	
1/bebo	bluecava	addvisor	Name, Zip code
1/bebo	bluecava	e.nexac	Name, Zip code
2/barnesandnoble	doubleclick	2mdn.net	Gender
1/gamespot	doubleclick	2mdn	Gender
2/youtube	doubleclick	googlesyndication	Gender
3/datehookup	doubleclick	pubmatic	IP Address
2/datehookup	doubleclick	criteo	IP Address
1/it.bab.la	adv.adsbwm	bid.openx	Ethnicity
1/travelocity	doubleclick	yieldmanager	Travel schedule
1/espncricinfo	doubleclick	2mdn	City
1/youtube	doubleclick	2mdn	Age, Gender
1/linkedin	doubleclick	2mdn	Zip code, Gender

As last experiment, we saw if users' habits influence NoTrace's effectiveness in reducing the data leakage, by simulating 100 different random navigation behaviors. Each navigation behavior has a navigation part and a Web search part. For the navigation part, we chose at random a set S consisting of 2 to 5 Alexa categories. For each category, we selected 5 random Web sites to log in and visit, while we visit the remaining 5 sites without signing in. For the Web search part we defined lists of popular search terms[14], one for each Alexa category defined in Section 4.1: each list will contain terms to search on `google.com` *relevant* to that category. Then, we searched on Google three terms, selected uniformly at random from the lists of popular terms relevant for each of the categories in S, therefore from 6 to 15 terms. Further 2 terms to search are chosen uniformly at random from the remaining lists, i.e., for the categories not in S. Results show that, regardless of the attitudes of the users while navigating the Web, the effectiveness of NoTrace is still high, as it effectively prevent any information leakage.

6. CONCLUSION

In this work we showed that NoTrace awareness empowers users with a clear overview of the availability of their PII, allowing them to make informed decisions about feasible privacy countermeasures. Moreover, NoTrace provides several measures to limit the diffusion of both personal and sensitive information, with higher efficacy and efficiency as compared to its most popular competitors. We also explored the most popular vectors for tracking, and how NoTrace is able to display these activities to users, and limit the diffusion of their private information. We showed that by reverse engineering what leakage is going to the top-10 aggregators, it is possible to discover what fraction of a user's profile is available to them. Our results show that one of the top-10 aggregator is able to collect 87% of a user's private information. Finally, unlike earlier work, we employed a crawling methodology that reflects users' real behaviors during online activities.

[14] `http://www.google.com/trends/explore`

7. REFERENCES

[1] C. Castelluccia, M.-A. Kaafar, and M.-D. Tran. Betrayed by Your Ads! In *PETS*, pages 1–17. 2012.

[2] G. Conti. *Googling Security: How Much Does Google Know About You?* Addison-Wesley, 2008.

[3] P. Eckersley. How Unique Is Your Web Browser? PETS '10.

[4] D. Galletta and et al. Web Site Delays: How Tolerant are Users? *JAIS*, 5(1), 2004.

[5] R. Gross and A. Acquisti. Information Revelation and Privacy in Online Social Networks. In *WPES*, 2005.

[6] B. Krishnamurthy, D. Malandrino, and C. E. Wills. Measuring privacy loss and the impact of privacy protection in web browsing. SOUPS '07, pages 52–63, 2007.

[7] B. Krishnamurthy, K. Naryshkin, and C. E. Wills. Privacy leakage vs. protection measures: the growing disconnect. In *W2SP*, 2011.

[8] B. Krishnamurthy and J. Rexford. *Web protocols and practice: HTTP/1.1, Networking protocols, caching, and traffic measurement*. Addison-Wesley, 2001.

[9] B. Krishnamurthy and C. Wills. Privacy diffusion on the web: a longitudinal perspective. In *WWW*, 2009.

[10] D. Malandrino and et al. Privacy Awareness about Information Leakage: Who knows what about me? Technical report, University of Salerno, 2013. `http://www.di.unisa.it/~delmal/papers/UNISA-ISIS-082913TR.pdf`.

[11] D. Malandrino and V. Scarano. Supportive, Comprehensive and Improved Privacy Protection for Web Browsing. In *PASSAT*, pages 1173–1176, 2011.

[12] D. Malandrino and V. Scarano. Privacy leakage on the Web: Diffusion and countermeasures. *Computer Networks*, 57(14):2833 – 2855, 2013.

[13] D. Malandrino, V. Scarano, and R. Spinelli. How increased awareness can impact attitudes and behaviors toward online privacy protection. In *PASSAT*, 2013.

[14] H. Mao, X. Shuai, and A. Kapadia. Loose Tweets: An Analysis of Privacy Leaks on Twitter. WPES '11, 2011.

[15] A. Narayanan and V. Shmatikov. De-anonymizing social networks. SP '09, pages 173–187, 2009.

[16] D. Perito and et al. How Unique and Traceable Are Usernames? In *PETS*. 2011.

[17] S. Pötzsch. Privacy Awareness: A Means to Solve the Privacy Paradox? In *IFIP AICT*, volume 298. 2009.