2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress

Plausible Deniability for ISP Log and Browser Suggestion Obfuscation with a Phrase Extractor on Potentially Open Text

Ronald Loui
Department of Computer Science
University of Illinois
Springfield, IL
r.p.loui-at-gmail

***Abstract*—We revisit the issue of maintaining a reasonable amount of privacy for browser users when logs are at risk of being sold and profiles are at risk of being viewed under embarrassing circumstances. A URL that generates dynamic calls in javascript is designed so users have plausible deniability when items show up in profiles, recommendations, or suggested text completions. Three design features are notable: (1) the use of open seed text and an unrestricted user-customizable list of URLs provided externally by the user, and not logged, so broad, indisputable claims of origination are possible; (2) the use of an AI/NLP part-of-speech informed phrase extractor (chunker) to generate queries that cluster semantically; (3) the use of a generative grammar of search-term sequences, or dialogues, to produce more realistic searches.**

An algorithmic technique is employed to assist in determining what requests were made by the tool post hoc, so forensics are still possible, but only when a specific user's logs have been produced for analysis. We also discuss how an "arms race" between analytics AI and obfuscating AI can be won by combining multiple tools, and how such a race can be avoided simply by deliberately reducing precision, unilaterally, by those who do user profiling analytics.

Keywords—browser log privacy, TRACKMENOT, obfuscation, plausible deniability, disinformation, concealing user behavior, ISP logs, user profiling, ad services, analytics, disinformation

I. Introduction

With the recent US government decision to permit ISPs to sell browsing information of users, there is a wider need for privacy enhancing tools that generate fake user activity, for obscuring real user activity. There is also an opportunity to revisit some of the decisions made by designers of earlier tools. Our interest in designing privacy-enhancing concealers started independently of this decision, but is perhaps a well calibrated response. Even before the call for obfuscation was made widely and coherently by Brunton and Nissenbaum [1], we were prototyping methods for users to have plausible deniability. Howe and Nissenbaum's TRACKMENOT [2] is the recognized pioneer in this area, dating from about 2006.

TRACKMENOT [2] was designed mostly in an attempt to cover user activity when privacy advocates were concerned with advertiser profiling, google profiles, and government surveillance by intelligence agencies. The aim was thus covering, or obfuscating, the user's behavior. TRACKMENOT's version 1 started with a URL that could be visited, that would generate fairly random queries. But websites could respond with programming on the client side of the browser that would send information about how the queries were generated. Information could include mouse and click dynamics, where a human-initiated request and simple *window.open()* call would clearly differ. The user thus had to be simulated. Version 2 of TRACKMENOT required the user to install a browser extension. This decreased usability and increased the trust required of third-party programming (it eventually led to tools like AdNauseum which google banned for attacking their revenue model directly).

"Polluting" a user profile and "trashing" logs at a vendor web site is, however, different from providing a user an effective level of deniability at the ISP logging and browser auto-completion levels. The former may encounter the adversary's ability to add programming to acquire more information about queries that are directed to their site. But the latter has a more comprehensive view of the user's behavior, across many sites. We propose that a less sophisticated approach to generating browser-based URL requests suffices for ISP log obfuscation. So we return to the first TRACKMENOT idea of generating activity from a web page rather than from a browser extension. However, with the more comprehensive view of activity, our efforts aim at confusing the user modeling, with stronger AI methods, rather than user interface simulation.

II. Blowing Smoke

The basic service architecture is to provide a web site that generates a web page with javascript calls to a set of URLs, and/or queries to search engines. Note that the queries are thus generated with the user's IP address and in the user's browser. The intellectual question is what queries to generate.

TRACKMENOT used "dynamic query lists" seeded with popular news RSS feeds such as NYT, CNN, and Slashdot. It added "a list of popular queries gathered from publicly available lists of recent search terms." Responses from requests have terms extracted for appending to the query list, based on regular expression pattern matching. This has lots of dynamism, but was criticized for not producing query lists that

978-1-5386-1956-8/17 $31.00 © 2017 IEEE
DOI 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.58

clustered the way that human queries would. It also restricted its activity to search engine queries and did not venture much beyond the terms appearing in the feed. Although a dynamic list is better than a static list, and a seed text that is somewhat coherent produces clusterable activity, one can do better.

1. Our landing page presents the user with a textbox for URLs and query terms that can be customized at will. If the user chooses to use one of the many default lists, there need not be customization. It is important that there is the possibility of the user providing external lists of seed text, in open, unrestricted and unlogged (by our site) form. This provides the deniability, because one *could have* generated a URL or query from the user's seed. Perhaps the probability is small, especially when looking at sequences of requests, but *no one could know for sure*, at least *prima facie* (see later remarks on forensics).

2. Seed text that is placed by the user in a text box is scanned for phrases that look like technical terms (the concept of technical phrase comes from a legal contract analysis project in AI and Law, from which this scanner is derived). This scanner is aware of some parts of speech, such as adjective, adverb, typical adverbial endings, connecting prepositions and conjunctions, stop words, non-ending words, and a target grammar. Many available chunkers do similar work, but our scanner is linear in time and space, so it can be applied liberally to custom user input; it also has good results when many novel words are not found in a part-of-speech dictionary. A user is free to provide anything with English or European terms and phrases. For the generation of default text, when customization is not chosen, we used wikipedia article text. A sampling of terms, including proper names, extracted from the wikipedia article on cyberspace is:

communicative resources
network of information
co-workers and providers
johann steininger
global communication network predicted
historical moment
society and culture
semantic meaning
physical-virtual dynamics
anniversary edition
wayback machine
journal of politics
relationship between different pages
in official governmental sources
anonymous ownership
computer-aided architectural design research
various thought experiments
remediation of culture
hierarchical ordering principle

Generating queries with these terms, by hypothesis, will be coherent because they came from a coherent seed text. Queries are especially coherent when they contain proper names related to a subject, as here from a wikipedia list of comedians:

little rascals
patricia heaton
valerie harper
carlos alazraqui
patrick warburton
felicia michaels
dieter hildebrandt
jasper carrott

3. A probabilistic generative grammar mixes these terms, so cross-domain, frequent modifiers are added, randomly, as prefixes or suffixes. For example, “2017” or “top ten” might modify the selected phrase. We plan in the future to create some of these modifiers based on actual user search query logs (from third party sites, not our own). Currently, there are 200+ kinds of modifications, some of which generate multiple possibilities, e.g., a year from 1990 to 2017. Since some of the queries are aimed at filling browser URL suggestion or auto-completion options, special attention is given to generating queries starting with less frequent letters, such as “x” and “z”. This mechanism is also capable of generating random misspellings and typo errors, at low rates, and dropping small words that a human might omit for brevity (deletions and permutations). Syntactic modification of URLs, resulting in deliberate 404 NOT FOUND errors, is a feature that decreases the burden on targeted servers.

A more important idea is a carried-concept, used across multiple searches as a suffix. This carried concept can be varied in its form (sometimes losing a word or two), but gives the impression of a search session that has a temporarily fixed target. For example, this sequence appears as query terms extracted from seed text, prior to the 200+ probabilistic modifications being applied (nonprofit organization is a carried concept):

funded campaign
fundraising campaign
crowdfunding website indiegogo
nonprofit organization
property in shoreham nonprofit organization
property and restore nonprofit organization
eventually building nonprofit organization
serbian-american electrical engineer nonprofit organization
engineer and inventor nonprofit organization
inventor nikola nonprofit organization

Coherence is supplied both by the seed text and the carried concept. Actual generated modifications in one chance embodiment included “funded campaign 1996,” “making fundraising campaign,” and just “nonprofit.”

Finally, queries are generated to multiple kinds of google search service, e.g., news, images, youtube, with various probabilities and with times varied over a user-selectable base rate (12 second default). Queries are issued with the meta-tag to suppress sending browser referral page information.

III. Real Effects and Forensic Assistance

The performance of this tool depends of course on the algorithms used to generate profiles, the browser settings and past user behavior, as well as the tactics for browser auto-completion and suggestion. What we have observed repeatedly, with many users over several months, are the following:

a. about half of the browser suggestions based on a single first character can quickly be replaced with open-auto-generated terms, after a single “session” with the tool; it is difficult to remove all of the human-generated auto-completions, especially after two or three characters.

b. about half of the youtube.com home page video suggestions can be replaced with suggestions originating from open-auto-generated queries and URLs; the queries are not as

effective as the URLs, and the appearance of watching a video to completion (with *&t=nnn* appendings to requests) is more effective than simply starting; after a day or two, the effect of the obfuscation can subside, even if the user does not create new human-generated queries (no doubt the user-profiling has a recency-weighting scheme).

c. some queries and suggested topics are more effective at creating deniability than others; for example, smart phone queries that were open-auto-generated are effective obfuscators for someone with tech interests; NHL hockey fights and videos with faces draw the eye away from actual suggested content that is less attention-grabbing, such as Cheese Making videos; youtube is a user profiling site, but it gives some insight into how the ISP log analytics might return results.

d. for a real user, 50% correctly profiled suggestions based on intentional behavior remains useful for increased efficiency, while providing cover and deniability.

IV. Forensics

There is a method for subtracting auto-generated requests, even when the seed lists and text are potentially open and are unlogged by us. If law enforcement were looking to prosecute criminals, for example, all they need to do is provide a complete log of the user's activity. Queries generated by this tool are separated in time according to a formula that is based on the characters in each subsequent query, or across small groups of queries. Knowing that formula, the entire set of queries can be removed, and the user's actual behavior can be revealed. However, this can only be done after logs have been produced by someone other than ourselves, and only by someone who can do the requisite reverse engineering. The formula is easy to change by day or hour. Privacy is protected by the difficulty of producing such logs, the legitimation of the request for the mathematical connection, and the effort to do the subtraction. In some ways, this enforces an individualized warrant and makes difficult any bulk surveillance.

V. Final Thoughts

One might ask whether this tool is aimed at breaking the ad revenue model or frustrating sites like Facebook and Google. The answer is no. Sites such as Facebook and Gmail can know a lot about a user that would require much more aggressive disinformation to frustrate. Many service providers need ad revenue to stay in business.

We are much more interested in ISP and browser logs, perfect data loss and theft, blackmail, making embarrassment less frequent, and providing deniability. It is our hope that ethical, intelligent providers like Google and Facebook will provide similar deniability, or partial concealment, for their users after seeing the value provided here. They could achieve this simply by adding about 33% fairly random suggestions whenever they make suggestions from a user profile, i.e., by being deliberately less accurate. No doubt reduced accuracy would also reduce some of their costs, because 70% good enough, which is very helpful, is cheap compared to near perfection, which helps no one.

There are still some risks to the user: If one provides a URL and the URL has session information, and one is still logged into an account, that could make the server do something unintended. This could also happen if a login is not required and can change state on the server. There are of course risks that a user will gain unintended attention because the user generated a false positive for some analytic algorithm. There is the risk of polluting one's own profile so the recommendations are not only obscured, but genuinely unhelpful. There is the risk that someone will not believe the person who claims that the activity was generated by an obfuscator. There is the risk that google and other web services will detect that one is using automatic queries and will take action to prove one is a human, or worse. There is the risk of generating unintended DDOS attacks on sites and placing demands on local devices. There is a risk that some will over-estimate what this tool can do, and will feel more secure than they should.

This is one further step in the development of privacy-enhancing browser automation. This tool, MYCONCEALER, aims at deniability with stronger AI methods, while returning TRACKMENOT to its original, simpler form. Log analytics could respond with improvements, though tool development on the user side could improve again. In some ways, the simplicity of ideas here is a deliberate feint to provoke an un-scalable adversarial response. In an arms race between AI on the user side and AI on the analytics side, probably enough doubt could be retained for deniability. A variety of obfuscators in tandem, with crowd-shared logs for URL lists, would obscure even more effectively. Complete self de-ideintification would be harder to achieve.

Acknowledgment

The Tech Brainstorming Club at UIS and several students in the Graduate Research Seminar provided useful ideas, especially S. Nandikanti, S.K. Paruchuri, J. Andersen, E. Syed, K. Kanwar, R. Doddi, S. Khedkar, S. Kulkarni, P. Joshi, and D. Srirangapalle.

References

[1] Brunton, Finn, and Helen Nissenbaum. *Obfuscation: A user's guide for privacy and protest*. MIT Press, 2015.

[2] Howe, Daniel C., and Helen Nissenbaum. "TrackMeNot: Resisting surveillance in web search." *Lessons from the Identity trail: Anonymity, privacy, and identity in a networked society* 23 (2009): 417-436.

[3] Chow, Richard, and Philippe Golle. "Faking contextual data for fun, profit, and privacy." *Proceedings of the 8th ACM Workshop on Privacy in the Electronic Society*. ACM, 2009.

[4] Peddinti, Sai, and Nitesh Saxena. "On the privacy of web search based on query obfuscation: a case study of TrackMeNot." *Privacy Enhancing Technologies*. Springer Berlin/Heidelberg, 2010.

[5] Toubiana, Vincent, Lakshminarayanan Subramanian, and Helen Nissenbaum. "Trackmenot: Enhancing the privacy of web search." *arXiv preprint arXiv:1109.4677* (2011).

[6] Al-Rfou, Rami, William Jannen, and Nikhil Patwardhan. "TrackMeNot-so-good-after-all." *arXiv preprint arXiv:1211.0320* (2012).

VI. APPENDIX

The two figures below show the landing page with customizable optional open URL list and open seed text, and the launch page, returned from the server, with javascript for user control at the item level. Queries are auto-generated by the methods described above: AI/NLP-extraction + concept-carrying + probabilistic-generative-grammar + timing and targeting.