A Promising Direction for Web Tracking Countermeasures

Jason Bau, Jonathan Mayer, Hristo Paskov and John C. Mitchell
Stanford University
Stanford, CA
{jbau, jmayer, hpaskov, jcm}@stanford.edu

***Abstract*—Web tracking continues to pose a vexing policy problem. Surveys have repeatedly demonstrated substantial consumer demand for control mechanisms, and policymakers worldwide have pressed for a Do Not Track system that effectuates user preferences. At present, however, consumers are left in the lurch: existing control mechanisms and countermeasures have spotty effectiveness and are difficult to use.**

We argue in this position paper that machine learning could enable tracking countermeasures that are effective and easy to use. Moreover, by distancing human expert judgments, machine learning approaches are both easy to maintain and palatable to browser vendors. We briefly explore some of the promise and challenges of machine learning for tracking countermeasures, and we close with preliminary results from a prototype implementation.

I. Motivation

It is a transformative time for web privacy.

Just a decade ago, web tracking received scant scrutiny from researchers, policymakers, and consumers. There were few third-party services following users about the web, they were concentrated almost exclusively in the behavioral advertising market, and they relied solely on easy-to-block identifier cookies.[1] In the intervening years, tracking has boomed: there are hundreds of companies, dozens of business models, and myriad tracking technologies in use [1].

Web users are concerned: surveys have consistently demonstrated that consumers want control over web tracking [2], [3], [4], [5], [6]. Not coincidentally, governments have taken steps to intervene. Beginning in 2010, policymakers in the United States [7], [8], Canada [9], and the European Union [10], [11], [12] issued vocal calls for new technical approaches to consumer choice about web tracking. The policy discourse has thus far largely centered on Do Not Track [13], an initiative that combines preference-signaling mechanisms with a substantive compliance policy and out-of-band regulatory enforcement.

As of early 2013, however, Do Not Track remains a nascent proposal. Companies and trade groups involved in web tracking have vigorously opposed the concept, and progress towards standardization in the World Wide Web Consortium has essentially stalled [14], [15].

While Do Not Track remains pending, there has been a renewal of interest in technical countermeasures against web tracking. Mozilla [16] and Microsoft [17], [18] have already shipped new anti-tracking features, and Apple has indicated openness to improving its longstanding countermeasures [19]. The chairman of the United States Federal Trade Commission recently departed from the agency's longstanding focus on Do Not Track and signaled that it views technical countermeasures as a promising direction for consumer control [14].

Moreover, even if Do Not Track is widely adopted, some websites may ignore or deceptively misinterpret the signal. Rigorous technical countermeasures would provide a necessary backstop for Do Not Track.

But there's a problem: existing technical countermeasures are grossly inadequate. Web browsers and extensions have long offered a range of web tracking countermeasures, representing a variety of approaches to tracking detection, mitigation, and accommodation. The most effective [1] solutions—manually-curated block lists of tracking content—are difficult to maintain and difficult to use. Meanwhile, the most usable [20] countermeasures—those built into web browsers—are among the least effective. Though several browser vendors seek to provide user control over web tracking, they are loathe to individually identify companies and invite the business and legal scrutiny of picking and choosing among marketplace participants. Browser vendors consequently rely on inaccurate heuristics to identify trackers (e.g. comparing domains). Owing to the prevalence of false positives, browser countermeasures are limited to small interventions (e.g. blocking cookies).

Machine learning charts a promising new course. It could provide the accuracy of a curated block list—allowing for rigorous privacy measures (e.g. HTTP blocking). And it distances expert human judgments from identification, reducing costs and potential risks and allowing usable implementations by browser vendors. We believe a machine learning approach that identifies tracking websites is now viable, and in Section II we sketch possible architectures and associated challenges. Section III closes with preliminary results from

[1] Detailed definitions of "third party" and "tracking" are hotly contested. For purposes of this position paper, we mean simply unaffiliated websites and the collection of a user's browsing history.

a prototype implementation.

II. Machine Learning

In this section, we briefly review the necessary design components for a machine-learning system that identifies web trackers. We discuss possible architectures for data collection and what data should be collected. We then turn to sources of training data and, finally, machine learning output.

A. Data Collection Architecture

There is a continuum of possible architectures for continually collecting the web measurements needed to identify trackers. At one end, a centralized crawler could periodically visit websites, much like modern approaches to web search. At the other end, a decentralized reporting system could collect information from users' browsers. A crawler has the advantage of easy implementation and disadvantages of possibly unrealistic browsing patterns and special-case behavior by websites. Crowdsourced data provides the benefit of closely modeling user experience, with drawbacks of privacy risks and potential gaming.

The two poles are hardly exclusive; a tracking detection system could make use of both crawled and crowdsourced data in varying degrees. We believe a centralized approach is most feasible in the near term, with gradual and careful expansion into decentralized collection.

B. What Data to Collect

At a high level of generality, there are two categories of data to collect.

First, information that reflects the relationships among web content. Effective tracking requires observing a user's behavior across multiple websites, and many trackers collaborate to share data (e.g. advertising exchanges). The Document Object Model (DOM) hierarchy provides a starting point, reflecting the context that web content is loaded into. The DOM's value is limited, however, in that its purpose is to reflect layout and security properties—-not provide privacy-relevant attribution about how content came to be loaded. A machine learning system would have to account for `script` embeds, redirects, and other dynamic mechanisms that introduce content into a webpage.

Second, information that reflects the properties of particular web content. Type, size, cache expiry, and many more features vary between trackers and non-trackers. Similarly, usage of various browser features differs by a website's line of business.

To illustrate the need for both categories of data, consider the case of a popular content distribution network (CDN) that does not engage in tracking [21]. A naive scheme that merely considers the prevalence of web content would identify the CDN as a tracker. A correct classification requires information about the CDN's behavior—for example, that it does not set cookies. Conversely, consider the case of a website that uses outsourced—but carefully siloed—analytics from a third-party domain. The analytics content may appear to be highly intrusive, setting unique cookies and inquiring about browser features. Without knowing that the analytics service is directly embedded by first parties and never draws in other third parties (Figure 1), it might be erroneously classified as tracking.

Some trackers may attempt to evade detection by tuning their behavior. Efforts at masking tracking are fundamentally limited, however, by the need to provide content on many websites, collaborate with other trackers, and invoke particular browser functionality. A technological cat-and-mouse game between new tracking techniques and detecting these techniques would also seem to favor detection, due to much less vested infrastructure. Furthermore, as we discuss in Section II-D, adjustments in practices to circumvent countermeasures may both be impractical and subject a business to media, business, and legal penalties.

Web browser instrumentation would be sufficient to capture both categories of data. The FourthParty platform [22] offers a first step, though modifications are necessary to properly attribute content that is inserted by a `script`.

C. Labeled Training Sets

Curated block lists are one possible source of training data. Lists vary in quality; some are very comprehensive [1].

Industry-provided lists are another potential source and possibly less objectionable for browser vendors. The online advertising industry's trade group, for example, maintains a list of member company domains.

D. Machine Learning Output

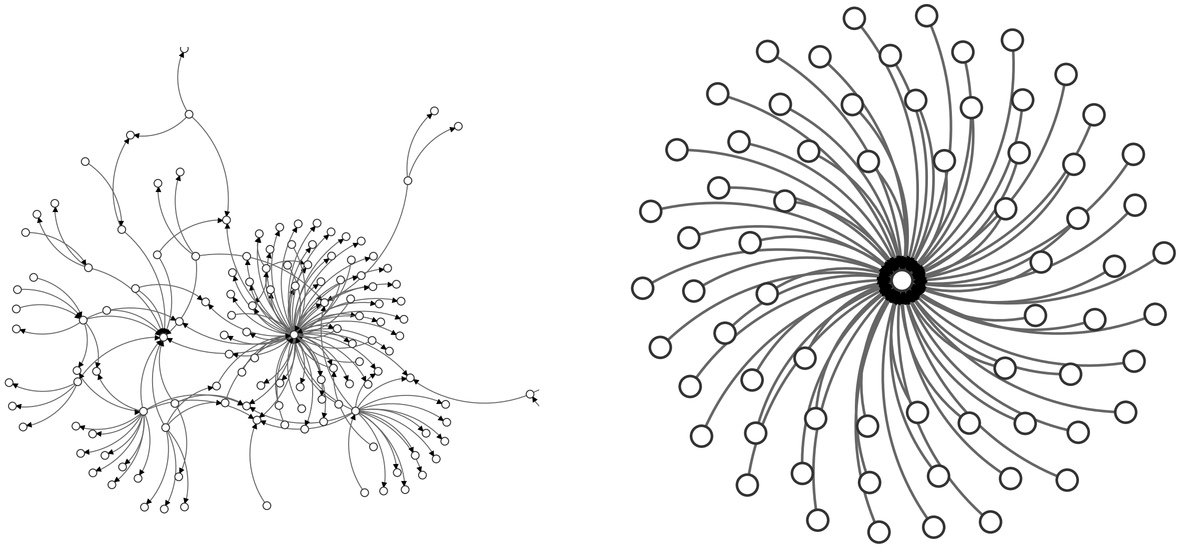
The machine-learning system must be able to provide a real-time judgment about whether particular web content is tracking the user. These determinations might be entirely precomputed and static, or they might be dynamically generated by a classifier within the browser. In our view, a static list is preferable in the near term owing to simpler deployment and compatibility with existing block list mechanisms.

We believe domain-level[2] granularity would be sufficient in these static lists, for the following reasons.

First, tracking often occurs on a dedicated domain. Exclusively third-party businesses usually only have the one domain. Firms that operate both first-party and third-party web services tend to separate those services by domain, reflecting path dependency from acquisitions, internal organizational boundaries, and security and privacy design considerations.

Second, the rate of change for tracking domains will be much slower than the rate of data collection and classifier training. Swapping domains involves substantial logistical

[2] In the following discussion, by "domain," we mean a public suffix + 1 [23].



(a) Script loading relationships involving the advertising domain

(b) Script loading relationships involving the analytics domain

Figure 1. Loading relationships between script source domains captured on a crawl of Top 3000 Alexa sites. Each circle represents an individual domain. Each arrow $A \rightarrow B$ indicates that the execution of a script from domain A caused a script from domain B to also be loaded into the DOM, on at least one of the crawled pages.

complexity; a website's partners and clients would all have to direct to the new domain. Moreover, shifting domains could cost a website its historical tracking data since the same-origin policy would prevent accessing old cookies and other stateful tracking technologies. Meanwhile, the shift in tracking domains would be observable by the detection infrastructure with little technical change or additional cost.

Third, when a tracking service is ostensibly hosted on a first-party subdomain, it might still be detected by inspecting DNS CNAME records.[3]

Finally, if a website intentionally uses domain name trickery to circumvent the system, it may face business, legal, and press repercussions. When businesses circumvented the Safari cookie blocking feature, for example, they were promptly lambasted in the media, sued, and in one instance steeply fined by the Federal Trade Commission [24].

The output of a machine-learning system could be connected to various technical limitations, ranging from restrictions on privacy-related browser APIs (e.g. cookie blocking) to entirely blocking HTTP traffic. A particularly promising direction is to scale the degree of technical limitation with the degree of confidence that a website is engaged in tracking. Erroneously classifying a CDN as a tracker, for example, could break a website if connected to HTTP blocking, but may only result in a minor performance degradation if connected to restricted storage access.

III. Preliminary Results

We collected the data for our initial experiment by crawling popular websites with FourthParty [22]. Our crawler visited the Quantcast United States top 32,000 homepages, then randomly followed 5 links on each page that shared the page's domain. We generated DOM-like hierarchies from the crawl data, with a tree rooted at each webpage that the crawler visited. Interior nodes and parent-child relationships reflected `iframes`; leaf nodes and parent-child relationships reflected all other web content. We labeled each node with its domain.

We calculated aggregate statistics for each domain based on the set of trees in which it appeared. These statistics included the minimum, median, and maximum of depth, occurrences, degree, siblings, children, unique parents, unique children, and unique appearances per tree. We then trained a variant of the Elastic Net algorithm [25] using these statistics as features. Training and testing labels were sourced from a popular block list.[4] We used an 80%-20% split for training and testing data and determined all tuning parameters for the algorithm through 10-fold cross-validation.

Figure 2 depicts the performance of our classifier by plotting the proportion of trackers correctly identified, i.e. precision, against the proportion of non-trackers erroneously labeled as trackers, i.e. false positive rate (FPR). We show performance over a range of FPRs because browser-based technical limitations vary in their tolerance for false pos-

[3] For example, `metrics.apple.com` CNAMEs to `appleglobal.112.2o7.net`. Including `2o7.net` in the machine-learning output would be sufficient for a proper classification.

[4] We manually edited the list to remove first-party domains and add missing third-party domains.

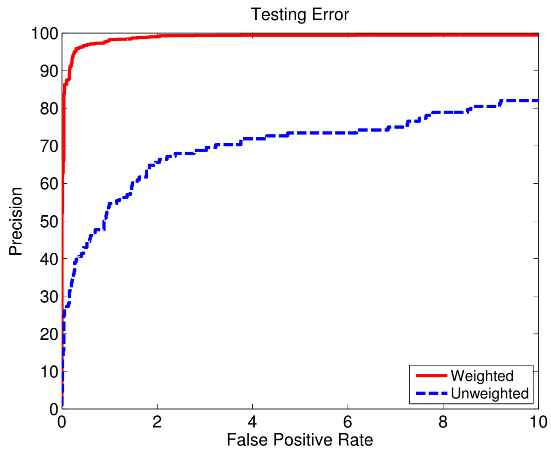


Figure 2. Percent of trackers correctly identified versus percent of misclassified non-trackers. The weighted precision is calculated by weighting each tracker's importance according to its prevalence in our crawl data.

itives. We also present a weighted precision curve that accounts for the prevalence of each tracker, defined as the number of distinct first-party domains on which a tracker domain appears. This weighting scheme reflects the intuition that a tracker appearing on a handful of sites poses a much lesser invasion of privacy than a tracker that can effectively determine a user's entire browsing history.

Results with this simple classifier are compelling: we achieve a weighted precision of 96.7% at 0.5% FPR and 98% precision at 1% FPR. While unweighted precision is considerably lower, achieving 43% and 54% precision at 0.5% and 1% FPR, respectively, we note that 63% of the trackers in our testing set appear 6 times or fewer. Thus, there are many infrequent trackers that pose significantly less of a privacy concern than the multitudes of trackers appearing on hundreds or thousands of sites. Our ability to achieve such high weighted precision at low FPRs with very few statistics indicates that tracking detection is well-suited to machine learning and that more expressive features may enable a compelling privacy tool. We are currently investigating rich-feature representations to better detect infrequent trackers and to raise weighted precision at FPRs of 0.1% and lower.

IV. Conclusion

Given encouraging preliminary results, we believe that a machine learning approach can enable accurate and usable web tracking countermeasures, with the promise of impartiality and robustness to both natural and adversarial changes in tracker behavior. We are working to mature the machine-learning prototype presented here into a tracking countermeasure suitable for widespread deployment. To this end, we are improving classification *accuracy* through more sophisticated algorithms and richer feature properties taken from DOM hierarchy topologies and HTTP content headers. We plan to address the *impartiality* of our approach by trying to reduce the usage of labelled data and investigating how the features selected by our algorithms map to intuitively objectionable tracking behavior. Finally, we are characterizing and improving the *robustness* of our approach, by assessing the kinds of errors made by the classifiers and how well they adapt to changes in tracker behavior over time.

References

[1] J. R. Mayer and J. C. Mitchell, “Third-party web tracking: Policy and technology,” in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, 2012.

[2] C. J. Hoofnagle, J. M. Urban, and S. Li, “Privacy and modern advertising,” October 2012.

[3] Pew Research Center. (2012, March) Search engine use 2012. [Online]. Available: http://pewinternet.org/Reports/2012/Search-Engine-Use-2012.aspx

[4] TRUSTe and Harris Interactive. (2011, July) Privacy and online behavioral advertising. [Online]. Available: http://truste.com/ad-privacy/TRUSTe-2011-Consumer-Behavioral-Advertising-Survey-Results.pdf

[5] Gallup. (2010, December) USA Today/Gallup poll. [Online]. Available: http://gallup.com/poll/File/145334/Internet_Ads_Dec_21_2010.pdf

[6] J. Turow, J. King, C. J. Hoofnagle, A. Bleakley, and M. Hennessy, “Americans reject tailored advertising and three activities that enable it,” September 2009.

[7] Executive Office of the President, “Consumer data privacy in a networked world,” February 2012. [Online]. Available: http://whitehouse.gov/sites/default/files/privacy-final.pdf

[8] Federal Trade Commission Staff, “Protecting consumer privacy in an era of rapid change,” December 2010. [Online]. Available: http://ftc.gov/os/2010/12/101201privacyreport.pdf

[9] Office of the Privacy Commissioner of Canada. (2012, June) Policy position on online behavioural advertising. [Online]. Available: http://www.priv.gc.ca/information/guide/2012/bg_ba_1206_e.asp

[10] N. Kroes, “Privacy online: USA jumps aboard the “Do-Not-Track” standard,” February 2012. [Online]. Available: http://blogs.ec.europa.eu/neelie-kroes/usa-do-not-track/

[11] ——, “Why we need a sound Do-Not-Track standard for privacy online,” January 2012. [Online]. Available: http://blogs.ec.europa.eu/neelie-kroes/donottrack/

[12] ——, “Online privacy - reinforcing trust and confidence,” June 2011. [Online]. Available: http://europa.eu/rapid/pressReleasesAction.do?reference=SPEECH/11/461

[13] World Wide Web Consortium. Tracking Protection Working Group. [Online]. Available: http://www.w3.org/2011/tracking-protection/

[14] N. Singer, “Do not track? advertisers say ‘don’t tread on us’,” *The New York Times*, October 2012.

[15] J. Blagdon, “Do not track: an uncertain future for the web’s most ambitious privacy initiative,” *The Verge*, October 2012.

[16] J. Mayer. (2013, February) The new firefox cookie policy. [Online]. Available: http://webpolicy.org/2013/02/22/the-new-firefox-cookie-policy/

[17] Tracking protection lists. [Online]. Available: http://ie.microsoft.com/testdrive/Browser/TrackingProtectionLists/faq.html

[18] Internet explorer 8 features: Inprivate. [Online]. Available: http://windows.microsoft.com/en-US/internet-explorer/products/ie-8/features/safer?tab=ie8inprivate

[19] (2012, June) Tracking Protection Working Group Bellevue face-to-face. [Online]. Available: http://www.w3.org/2012/06/20-dnt-minutes

[20] P. G. Leon, B. Ur, R. Balebako, L. F. Cranor, R. Shay, and Y. Wang, “Why Johnny can’t opt out: A usability evaluation of tools to limit online behavioral advertising,” Carnegie Mellon CyLab, Tech. Rep. 11-017, October 2011.

[21] N. Nikiforakis, L. Invernizzi, A. Kapravelos, S. Van Acker, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, “You are what you include: Large-scale evaluation of remote javascript inclusions,” in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, Raleigh, NC, October 2012.

[22] FourthParty. [Online]. Available: http://fourthparty.info/

[23] Mozilla Foundation. Public suffix list. [Online]. Available: http://publicsuffix.org/

[24] “Google fined over Safari cookie privacy row,” http://www.bbc.co.uk/news/technology-19200279, August 9 2012.

[25] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301–320, 2005.