Engineering Privacy and Protest: a Case Study of AdNauseam

Daniel C. Howe
School of Creative Media
City University, Hong Kong
Email: daniel@rednoise.org

Helen Nissenbaum
Cornell Tech
New York University
Email: hfn1@nyu.edu

Abstract—The strategy of obfuscation has been broadly applied—in search, location tracking, private communication, anonymity—and, as such, has been recognized as an important element of the privacy engineer's toolbox. However, there remains a need for clearly articulated case studies describing not only the engineering of obfuscation mechanisms but, further, providing a critical appraisal of obfuscation's fit for specific socio-technical applications. This is the aim of our paper, which presents our experiences designing and implementing AdNauseam, an open-source browser extension that leverages obfuscation to counter tracking by online advertisers.

At its core, AdNauseam works like a list-based blocker, hiding ads and blocking trackers. However, it provides two additional features. First, it collects each ad that it finds in its 'Vault', allowing users to interactively explore the ads they have been served, and providing insight into the algorithmic profiles created by advertising networks. Second, AdNauseam simulates clicks on ads in order to confuse trackers and diminish the value of aggregated tracking data.

A critic might ask: why click? Why not simply hide ads from users and hide users from trackers? The twofold answer reveals what may be distinctive elements of the AdNauseam approach. To begin, we conceptualize privacy as a societal value. Whereas many privacy tools offer solutions only for individual users, AdNauseam is built on the assumption that, often, effective privacy protection must be infused throughout a system. This assumption presents different and interesting engineering challenges. Second, AdNauseam seeks to concurrently achieve the goal of resistance through protest. And since protest frequently involves being vocal, AdNauseam's core design at times conflicts with privacy conceptions based on secrecy and concealment. While such tensions, and the tradeoffs that result, are not uncommon in privacy engineering (for example, in private communications tools, between security properties and usability), the process of designing and building AdNauseam demanded their systematic consideration and resolution.

In this paper we describe the goals of the project, with a focus on the operational definition of privacy that serves as its guide, and the implemented features to which these goals map. We present the engineering approach employed, including tensions we encountered during implementation; how they were resolved and our methods of evaluation on both technical and ethical dimensions. We discuss challenges of distribution, including Google's ban of AdNauseam from its Chrome store. We conclude with thoughts on the broader challenges that face privacy tools which must operate within complex socio-technical contexts, especially those dominated by actors openly resistant to them.

I. Introduction

The ad blocking wars [35] reflect wide ranging resistance to aspects of the online advertising landscape, specifically tracking, malware, annoyance, degradation of browser performance, and bandwidth costs. A number of technical systems, utilizing diverse techniques, have addressed these issues in various combinations. AdNauseam, an open-source, cross-platform browser extension, contributes to this growing arsenal not only by hiding ads and blocking malware, but by leveraging obfuscation to confound those seeking to profile users based on interests revealed through ad clicks. Specifically AdNauseam hides online ads while clicking each ad behind the scenes in order to register visits in ad network databases. The goal of the software is to pollute the data gathered by trackers and, by diminishing confidence in this particular indicator, render their efforts to profile less effective. At the same time, the software allows users to engage in a form of expression, by actively disrupting the economic system that drives surreptitious tracking, and by creating mistrust (advertisers generally pay ad networks for clicks) within the advertising system. Additionally, AdNauseam makes the ads it collects available to users to explore at their convenience, via multiple interactive displays designed to facilitate real-time understanding of the advertising system at work.

A. Engineering Philosophy

Our approach to the development of AdNauseam builds on prior work that has explicitly taken social values into consideration during tool design [15], [13], [27]. Throughout planning, development, and testing phases, we have integrated values-oriented concerns as first-order "constraints" together with more typical engineering metrics such as efficiency, speed, and robustness. Specific instances of values-oriented constraints include *visibility and transparency* in interface, function, code, process, and strategy; *personal autonomy*, where users need not rely on third parties; *social protection of privacy* with distributed/community-oriented action; *minimal resource consumption* (cognitive, bandwidth, client and server processing); and *usability* (size, speed, configurability, ease-of-use). Enumerating values-oriented constraints early in the design process enables us to iteratively revisit and refine them in the context of specific technical decisions. Where relevant in the following sections, we discuss ways in which AdNauseam benefited from this values-oriented approach, as well as tensions between design goals that emerged. Additionally we have followed a number of strategies from the literature on privacy-by-design [20], [25], [21], [23], [6] by including *Data Minimization Strategies*, *Legitimacy Analysis* and *Socially-informed Risk Analysis* as elements of our design process.

B. Design Goals and Constraints

The AdNauseam extension is designed to realize three tangible goals for users, each of which addresses privacy in the context of online tracking via advertising. The first is to offer *protection*; protection against malware and "malvertising" (malicious software that

leverages advertising mechanisms to gain access to users' systems [32]), as well as protection against data aggregation and data profiling via clicks on advertisements. The second goal is to provide a means of proactive engagement, allowing users an avenue for *expression* of their dissatisfaction with current advertising mechanisms to those running the systems. In the case of AdNauseam, this expression has an interventional aspect, as the software actively attempts to disrupt the economic model that drives advertising surveillance. The third goal is to facilitate increased *understanding* of the complex advertising ecosystem—and the profiling on which it operates—by providing users with the ability to view the ads they are served in real-time, and later, to explore interactive visualizations of the ads collected over time. These mechanisms are augmented by in-interface links to additional learning resources.

C. Feature Mapping

The mapping of goals to features (and to the system modules described below) was performed as follows: The goal of *protection* was implemented at a basic level by the clicking of collected ads, via the visitation module; and by the blocking of non-visual trackers and malware, via the detection module. The former attempts to protect the user from data profiling via advertisements, and the latter from non-visual tracking and potential malware. *Expression* was realized through clicks, again via the visitation module, and also in our implementation of the EFF's Do Not Track (DNT) mechanism [12]. With DNT enabled (the default setting), the DNT header is sent with all requests, and ads on DNT sites remain visible. Ads on these DNT pages are also not visited by AdNauseam. The goal of increased *understanding* is realized via in the visualization module, specifically via the real-time menu interface, where users can watch as new ads are discovered, then visited; the vault interface (descibed below), and a range of explanatory links embedded throughout AdNauseam's settings pages. Additionally, an in-depth Frequently-Asked-Questions (FAQ) list is linked from multiple locations within the interface, leading to over 40 carefully answered questions.

D. Data Minimization

Following a growing body of literature on privacy-by-design [20], [25], [21], [23], [6], our design and implementation process followed principles of data minimization. Thus AdNauseam was designed to function without ever communicating to a "home server" or sending user-data to any other entity, for any reason. For developers, this meant we were unable to access usage patterns and related data, which may have yielded important insights. Yet this both clarified our own position in regard to data collection, and also enabled us to sidestep potential problems of data leakage or interception in transit. Similarly, this principle helped us to identify three pieces of private information which could potentially be leaked during normal function of the tool, and to then expose these as user options, with defaults that would provide normal users with full protection, while allowing advanced users the option to change these settings for higher performance. These data were all potentially exposed to third-parties (websites and advertising networks) as part of the normal HTTP headers sent with requests for ads being clicked; specifically page referer, user-agent, and cookies. From either the referer or cookies, an advertising network present on multiple sites sites (Google, for example, is estimated to have code running on 70% of websites [10]) could potentially recreate large portions of a user's click path, while the user-agent header could be used by malicious trackers attempting to perform browser fingerprinting [36]. Thus all such information was stripped from outgoing requests by default. As discussed below, this choice is not without consequences. Choosing to foreground the safety of user data in this way impacted AdNauseam's efficacy on other goals.

> *The application of data minimization does not necessarily imply anonymity, but may also be achieved by means of concealing information related to identifiable individuals* [20].

Additionally we have applied the principle of data minimization to our ad export feature, which allows users to export their aggregate ad collection (as JSON) in order to sync or migrate between machines, to backup, or to share. From our experiences user-testing this feature, we noted that such exports contained potentially sensitive data, specifically users' click trails (stored as the locations for found ads), possibly over months or even years. When considering how to handle this data we noted that it also existed in the browser's local storage, which could potentially be accessed by a malicious actor. Thus we subsequently implemented encryption for this data, both in memory and in storage, as well as offering the user the option, before each export, to redact all ad locations if desired.

E. Legitimacy Analysis

> *Before any privacy-by-design activities are embarked upon, a discussion needs to take place with respect to the "legitimacy" of the desired system given its burden on privacy. Legitimacy is described as "the establishment that the application goals would be useful for the intended use population."*[29][21]

A critic might ask: why click? Why not simply hide ads from users and hide users from trackers? There are two reasons. First, AdNauseam is inspired by the path-breaking work of Priscilla Regan, who argue that beyond its protection of individual interests, privacy serves social ends, much like a collective good such as clean air or national defense [38]. This commitment to privacy as a collective good presents interesting engineering and evaluation challenges, which, in our view, warrant close attention. Thus AdNauseam may stimulate deliberation not only on its particular features, but may draw attention to the conception of privacy it seeks to promote. A second reason for clicking, as opposed to simply blocking, is that AdNauseam seeks concurrently to achieve the goal of expressive resistance to tracking through protest. And since protest generally involves being vocal, AdNauseam's design seeks to give voice to users. Rather than promoting the conception of privacy as concealment AdNauseam provides a means for users to express, in plain sight, their dissent by disrupting the dominant model of commercial surveillance.

Lastly, critics have claime that AdNauseam harms "independent content producers who can no longer support their sites." As this critique touches a broad array of tools, including standard ad blockers, it will take us too far afield to address it in full here. However, setting aside the rejoinder which points out that these sites are enabling surveillance, or more harshly, "selling out" their visitors, the hope is that loosening the chokehold of tracking over web and mobile domains will allow other business models to flourish. Toward this end we have enabled support in AdNauseam for the EFF's DNT mechanism, a machine-verifiable (and potentially legally-binding) assertion on the part of sites that commit to privacy-respecting behavior [12]. For sites that make this commitment, AdNauseam does not (by default) hide, block, or click their ads.

F. Socially-informed Risk Analysis

Given the goals we hoped to achieve and the set of features to which these mapped, we set out to identify risks to which users might be exposed. For each such risk, we considered the degree to which the user would be exposed when browsing the web using an unmodified browser, in comparison to the degree of exposure while using AdNauseam. Finally we considered their exposure using existing alternatives, ad-blockers like AdBlock Plus [1] or wide-spectrum blockers like uBlock [18](see, for example, Figure 4 below). The following risks were identified: increased tracking by advertisers, leakage of personal data; and harm via malware/malvertising.

To establish a lower-bound on exposure, we set a constraint that user exposure with AdNauseam must be strictly lower on all dimensions than with our baseline case of browsing with an unmodified browser. Conversely, we hypothesized that the current performance of uBlock, the open-source blocker with the best performance metrics, would provide an upper-bound on risk exposure for the individual user. As AdNauseam must interact, at least minimally, with advertising servers in order to fulfill its functional requirements, it would necessarily expose users to more risk than the current state-of-the art blocker (e.g., uBlock). Notice that we refer specifically here to risk for *the individual*, a distinction we return to below. In all cases noted above (see *Comparative Evaluation* below), we were able to verify that risk to users was diminished with AdNauseam, both in comparison with the no-blocker case, *and* to AdBlock Plus, the most commonly installed blocker[37].

G. Design Tensions

1) Indistinguishability and Data Leakage: For obfuscation to function effectively as a means of counter-surveillance, the noise generated must exhibit a high degree of *indistinguishability* with regards to data the system intends to capture; that is, it must be difficult for an adversary to distinguish injected noise from the data it is attempting to collect [16], [3]. Thus we strive to make AdNauseam visit requests indistinguishable from those sent for manual requests. However, there are cases where this goal comes into tension with other aims of the software, specifically that of protection (both from malware and from data leakage). For example, when crafting a visit request, we must decide how accurately to mimic the data that the browser normally sends, specifically regarding user-agent and referer headers, and whether to include the cookies normally sent with the request. If we match these exactly then our request is indeed indistinguishable (assuming the browser is not modified to hide such details), and thus more difficult for an adversary to filter. However, this may also leak information to advertising networks that a user considers private, e.g., the URL of the page the ad was found on, the type of browser, operating system, etc. they are using. Similarly, we must decide whether to block incoming cookies from AdNauseam visits, whether the DOM of incoming request should be parsed and executed, and whether client-side scripts should be allowed to run.

While we can provide user configurable settings for all these parameters, we must still, for each case, consider the appropriate defaults by weighing risks to the user against potential gains in obfuscatory power (see *Risk Analysis*). For AdNauseam, we have decided on default setting that maximize user protection. Thus, by default we block the user-agent header (to deter browser fingerprinting) and the referer (so as not to leak page-view data). Similarly, incoming cookies from visits are blocked—though this should not have any direct effect on indistinguishability—further minimizing the tracking to which users are exposed. Because visits are implemented via AJAX requests, a DOM is not constructed from the response and client-side scripts are not executed. While protection is maximized here, obfuscatory power may be diminished. For example, one vector of attack we have noted is from an adversary who, upon receiving a click request, sends a preliminary response containing JavaScript code that executes from within the DOM. In this case, if the client-side script never runs, then the click is never executed. We have experimented with solutions that address this issue and thus better support indistinguishability (including executing clicks in sandboxed invisible tabs), but have yet to find a solution that adequately protects user data and is cross-platform. For the moment we leave this as future work.

2) Expression and Protection: We have talked of the expressive and protective capabilities of data obfuscation generally, and of AdNauseam specifically. Abstractly conceived, these two capabilities seem to lie at opposite ends of a spectrum—on the one end, providing protection against surveillance and profiling, on the other, enabling users to be heard in their expressions of critique or protest. On the protective end, the noise generated by a system must be indistinguishable. However, a tool that is perfectly protective—for which it is never possible to filter noise from data–will often be functionally invisible to an adversary. If an adversary is literally unaware of injected noise, then the expressive capability of that action is minimal (at least when considered from the perspective of the adversary). Conversely, if a system is highly expressive, it may be easier for an adversary to filter its noise, thus diminishing its protective capabilities. In the case of ScareMail (see *Related Work* below), it would be trivial for an engaged adversary to both detect users of the tool and to filter out noise generated by it. In practice, however, the relationship between expression and protection is more complex. Filtering might, in some cases, even serve to create temporary surveillance-free spaces. For example, an adversary filtering (and ignoring) data from ScareMail signatures might create a zone for messaging not subject to monitoring, at least temporarily. A variation on this dynamic may affect AdNauseam users. Were an ad-network capable and willing to react by filtering AdNauseam clicks, users would be in the interesting position of being ignored by the system they are trying to evade.

The case of TrackMeNot [27], a tool that obfuscates users' interests in web search, also embodies subtle interactions between protection and expression. It is relatively easy for search engines to detect usage of the tool by referring due to the sudden spike in search frequency due to noise queries. Here, regardless of detection or filtering, users have already successfully communicated their discontent, possibly even forcing search engines to address the increase in resource usage. However, even if search engines know that users are employing TrackMeNot, they may not be able to perfectly distinguish between "real" and "fake" queries; that is, there may be some percentage of generated queries that are indistinguishable from user-generated queries (see [16] for an analysis of this question). As such, the tool provides expressiveness as well as a degree of protection. An adversary may be aware that a tool is injecting noise into its system, yet be technically, culturally, or otherwise unable or unwilling to filter it. From a tactical perspective, as tool designers leveraging obfuscation, this combination of protection and expression may be a sweet spot toward which we aim. To do so however will require more precise definitions of these criteria—how can we more formally evaluate, for example, the expressivity of a system that embeds complex interactions between human and non-human actors? We return to this question in our conclusions below.

II. Architecture

The AdNauseam software is comprised of four modules, each responsible for one of its primary functions: detection, extraction, visualization, and visitation.

A. Detection

This module is responsible for the analysis and categorization of requests following a page view. Such requests, most often to third-parties, is first classified according to the type of elements it realizes; whether advertisements, analytics, beacons, social-media, or functional widgets. The largest proportion of such requests (40-50%) are made to the first group, on which this module focuses, which includes ad and ad tracking services [43]. This module determines which requests to block and which to allow, and, in the latter category, between those that yield visual elements and those used only for tracking.

In order to categorize such requests, we leverage the capabilities of the open-source uBlock-Origin [18] project, a configurable, list-based "blocker" that is effective and efficient [43]. Like other blockers, uBlock allows users to specify publicly accessible lists of resources which contain syntactic matching rules for the retrieval of web resources. Based on these lists, we first determine whether a request should be blocked or allowed, and then, if allowed, whether it should be visible or hidden. If hidden, the element is downloaded and included in the page, but made invisible to the user via a content-script. Both blocking and hiding are specified via rules that may include the serving domain, the type of resource (e.g., images or video), and/or properties of the DOM container (for example, a DIV with a specific id or class). Rules are included from widely distributed lists that are updated and maintained by individuals and communities (e.g, "EasyList" [9]. Additionally, users can augment these lists with custom rules they create, either to block or hide new content, or to whitelist a page, site or element.

Requests marked as blockable in AdNauseam are disallowed at the network level, mimicing the behavior of most other blockers, including uBlock, AdBlock Plus, Adblock, and Adguard, which perform blocking on some percentage of requests and hiding on some percentage of the remainder. The difference for AdNauseam is that a subset of requests which might be blocked in other blockers must be allowed in AdNauseam; specifically those that result in, directly or indirectly, visual advertisements.[1] At the element hiding level, the detection module is invoked incrementally, via content-scripts, as page elements are loaded (or dynamically generated) and inserted into the DOM. Elements marked for hiding are assigned a CSS class that sets their display to invisible, and the surrounding DOM is collapsed so as not to leave blank space on the page. Each hidden element (generally a visual ad) is then passed to the *Extraction* module.

B. Extraction

Once a visual element has been detected and hidden, we must then determine whether it is in fact an advertisement. If so, the extraction module of the system must extract the properties needed by the *Visualization* and *Visitation* modules. These properties include timestamp, size, content-url, target-url, page-detected-on, etc. Text-only ads, as often found on search engines, present a different challenge, as these are generally served inline along with page content rather than requested from a 3rd-party server. In these non-image cases, several additional fields are aggregated to form the content payload (title, description, tagline) and there is no content-url linking to an external resource. To enable extraction of such data, AdNauseam ships with a custom set of CSS selectors used to parse specific DOM attributes from text-ad sites (Google, Ask, Bing, etc.). Such filters run only on specific domains where text-ads have been previously discovered.

[1] Interestingly, it is exactly this standard combination of functions—hiding and blocking—that Google cites as being in violation of its Terms of Service, a claim discussed below in the *Distribution* section.

C. Visualization

In order to facilitate understanding of online advertising systems, AdNauseam provides users with interactive visualizations of their collected ad data. These visualizations provide both high-level displays of aggregate data (see Figure 1), as well as the option to inspect individual ads (see Figure 2), for data including the page on which the ad was found, the target URL, the text copy, the viewed-on date, ad network, and image or video 'content'. Additionally, a number of derived functions provide additional metrics (i.e., the total estimated charge to advertising networks for the ads visited on a given page or in a given time-period, as in Figure 3). Ads may be filtered and sorted according to a variety of criteria: by date, topic-category, ad-network, page-category, etc. The visualization module is a distinct contribution of AdNauseam, furthering our goal of increased understanding in two ways: 1) to enhance the user-experience with greater insight into the online advertising landscape; and 2) to enable interested users and researchers to study the ad data, and generate insight into the larger picture beyond momentary interactions. To facilitate these goals, we include mechanisms for importing and exporting ad data sets, which can be loaded and saved as plain-text JSON files directly from the extension. The use of this data, aggregated across users, for further research (with appropriate mechanisms for user consent) is an area we hope to explore in future work.

D. Visitation

This module simulates clicks (or visits) on collected ads, with the intention of appearing to the serving website (and ad network) as if the ad had been manually clicked. Currently, these clicks are implemented via AJAX, which simulates requests (matching headers, referer, etc.) that the browser would normally send. This provides users with protection against potential malware in ad payloads, as responses are not executed in the browser, and JavaScript, Flash, and other client-side scripting mechanisms are not executed. Similarly, AdNauseam blocks incoming cookies for responses to ad visits. In making these design choices, we favored user-protection (from potentially dangerous ad payloads) to the appearance of authenticity in AdNauseam clicks, as discussed in *Design Tensions* above.

What are the expected results of visiting some (or all) of users' collected ads? First, the data profiles of users stored by advertising networks and data brokers may be polluted, as users' actual interests are hidden by generated clicks. This both protects the individial user (assuming they have clicked or may click some ad in the future) as well as the larger user community, as aggregate statistics become less accurate. Second, as advertisers must now potentially pay publishers for decoy clicks, a degree of mistrust is introduced into the economic system. This is perhaps the most compelling argument for this strategy, as it could, given adequate adoption, force advertisers to change their behavior, either by developing new algorithms to filter such clicks, and/or by adopting more privacy-friendly policies (e.g., the EFF's DNT mechanism).

E. Distribution

Although not often discussed in an engineering context, issues surrounding the distribution of AdNauseam highlight concerns we

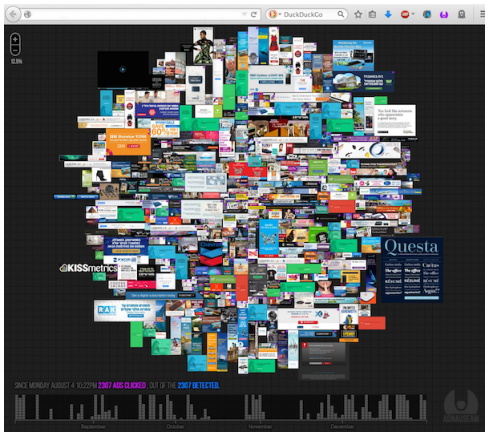

Fig. 1. AdNauseam's AdVault visualization.

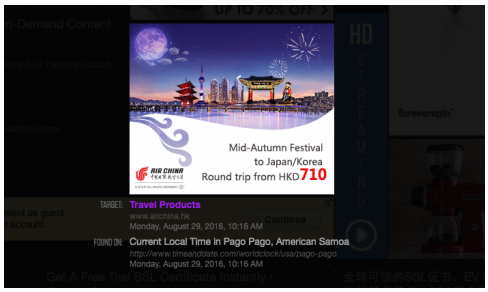

Fig. 2. Inspecting a single ad in the AdVault.

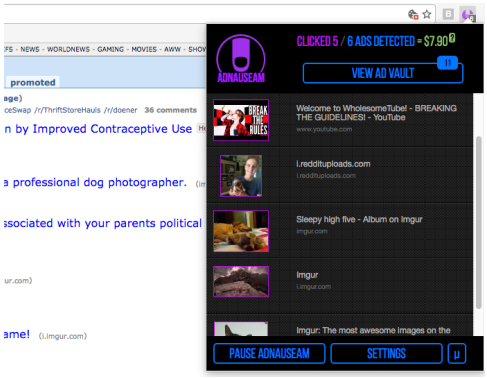

Fig. 3. Estimated cost to advertising networks.

imagine will become only more relevant as corporate players exert a growing influence over the software ecosystem.

The prototype for AdNauseam was initially developed as a Firefox-only extension available in Mozilla's addon store. In our production release, we added Opera and Chrome support and made the extension available in the Opera and Chrome stores respectively. We distributed upwards of 50,000 copies of the software over the subsequent six months, with the majority via Google's Chrome store. On January 1, 2017 however, we learned that Google had banned AdNauseam from the store, and further, had begun disallowing even manual installation or updates, effectively locking users out of their own saved data, all without prior notice or warning.

We wrote to Google requesting justification for the removal, and they responded that AdNauseam had breached the Store's Terms of Service, stating: "An extension should have a single purpose that is clear to users."[2] The single purpose of AdNauseam, we would argue, is not unclear at all—namely to resist the non-consensual surveillance conducted by advertising networks, of which Google is a prime example. Of course we understand that Google would prefer users not to install AdNauseam, as it opposes their core business model, but the Terms of Service do not (at least thus far) require extensions to endorse Google's business model. Moreover, this is not the justification cited for the removal. Whether or not one is an advocate of obfuscation, it is disconcerting to know that Google can quietly make a privacy extension, along with stored data and preferences, disappear without warning. In this instance it a counter-surveillance tool that is banned. Perhaps tomorrow it will be a secure chat app, or password manager. For developers, who, incidentally, must pay a fee to post items in the Chrome store, this should be cause for concern. Not only can one's software be banned without warning, but all comments, ratings, reviews, releases and statistics are deleted as well.

III. Comparative Evaluation

Qualitative evaluation was performed iteratively throughout development, often guided by solicited and unsolicited feedback from various constituencies, including users, developers, reviewers at Mozilla and Opera, and a range of privacy and security advocates. When considering how to evaluate the software, the question of whether AdNauseam in fact "worked" seemed at first to be most obvious and simple to address. We soon realized, however, that the meaning of this question shifted as users' goals, expectations, and perceived risks varied. Evaluating AdNauseam on the basis of feedback from the various constituencies was often a two-part process: first determining user orientations, and then examining feedback in light of their goals, concerns, and priorities. Additionally, beyond the technical issues with which we grappled, a subset of critiques consistently addressed ethical concerns. Thus we have split the discussion below into technical and ethical components.

A. Technical

Evaluation of obfuscation-based strategies for counter-surveillance is often a relatively straightforward challenge. As an example, consider the case of obfuscation in web search. To begin one might extract logs of queries from users, which contain both true queries and software-generated queries and run best-practice machine-learning (or other) algorithms to attempt to separate the two sets. Although one may not know the exact capabilities of an adversary (especially one like Google, with vast resoures at hand), one can make educated guesses as to the type of attacks that might be performed, whether, for the search case, based on timing, query content, side-channels, or other means (for details of evaluations in the search case, see [16], [3]). If we find that the adversary can differentiate true queries with high accuracy, then generated queries can be easily filtered and, from a protection standpoint, the tool has failed.

At first glance, evaluating AdNauseam seems similarly straightforward: to what degree of accuracy can an adversary, using state-of-the-art techniques, distinguish user clicks from generated clicks? The question is confounded however by the fact that for most users (those who enable hiding and disable DNT whitelisting), there are no true clicks to distinguish. No ads are seen, and thus none are ever clicked.

[2]In the one subsequent email we received, a Google representative stated that a single extension should not perform "both blocking and hiding," an assertion that is difficult to accept at face value as nearly all ad blockers (including uBlock, AdBlock Plus, Adguard, etc.) perform blocking and hiding, and have not thus far been banned.

Yet due to the frequency of clicks issued, an adversary will be able to readily ascertain that a user has installed AdNauseam. Given this information, we must ask what avenues are open to the adversary, whether technical or otherwise, and given these, what evaluation metrics might be applicable? One option is that data for the user in question is simply discarded. From the user's perspective this may be considered a win, as, at least in terms of clicks, they are no longer profiled. However, since the data is discarded, there is also no net gain in privacy when considered from a communal perspective.

But what if the user does not hide all ads? Here we must consider users who disable hiding altogether, as well as those who select a subset of ads to see, whether by manually whitelisting pages or by choosing to view ads on DNT sites. We imagine the former case—where a user goes through the trouble to install an blocker, but continues to view all ads– to be vanishingly small, and so focus on the latter, which consists of two sub-cases: the DNT case, and the non-DNT case. As DNT sites do not track users, protection is not needed, and generated clicks are not issued. In the whitelisting case, we find the situation that most closely resembles web search, where some subset of clicks are true and some generated. However the scenario is still different as here *all* ads are clicked (the corollary would be if a user could issue all possible queries to a search engine). Thus for an adversary to distinguish true clicks, there must be some element of the click request itself on which to filter, so we must verify that our requests, including HTTP headers, cookies and other data, are identical to manual requests (we check these daily via an automated testing framework). Even if requests are indistinguishable, however, there may still be side-channels available to more determined adversaries, as discussed in *indistinguishability* above.

1) Comparative: To further evaluate performance we compare AdNausam on with other commonly used blockers on a range of dimensions. Tests were first run without any extension, then with AdNauseam, Adblock Plus [1], uBlock-Origin [18], and Privacy Badger [11]. Tests were performed with each extension's default settings after resetting the browser to its install state. After visiting the websites in the test set (between 15 and 85 popular URLS, depending on the test) via the Selenium browser automation tool, we evaluated the safety of each extension in terms of the number of 3rd parties contaced (Figure 4), memory efficiency (Figure 5), and page-load speed (Figure 6). As shown in the graphs below, AdNauseam performed better on all dimensions than no blocker and, perhaps surprisingly, AdBlock Plus. As expected, AdNauseam performed less well than uBlock, due to the need to allow visual ad resources, rather than blocking them outright. Privacy Badger varied according to the test in question and on whether it had been pre-trained.

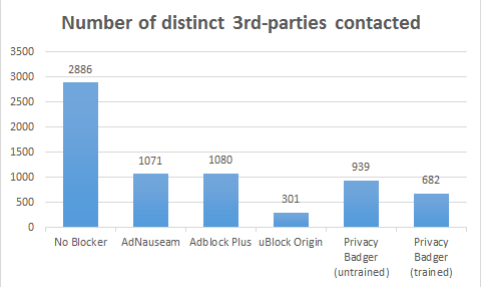

Fig. 4. Number of distinct third-parties contacted.

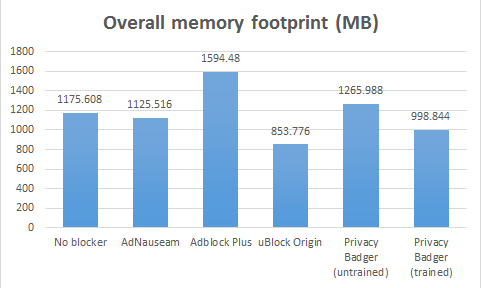

Fig. 5. Overall memory footprint (MB).

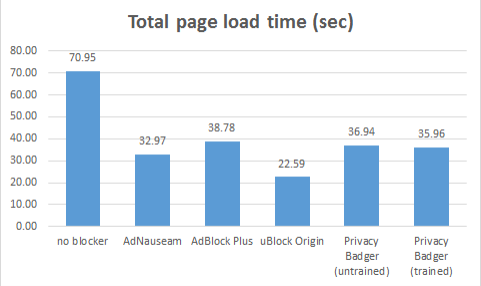

Fig. 6. Total page load time (sec).

B. Ethical

In adopting the philosophy of data obfuscation AdNauseam seeks to shields users from the inexorable, and inappropriate probes of services and third parties. Choosing obfuscation, however, means taking seriously the ethical critiques that it has drawn, including charges that it is dishonest, wastes resources, and pollutes data repositories. Addressing these issues in *Obfuscation: A User's Guide to Privacy and Protest* [5], the authors charge creators of obfuscating systems to answer two questions: first, whether their aims are laudable; and second, whether alternative approaches exist which might achieve these aims at lesser cost. Regarding the first charge we take as a point of departure that ubiquitous online surveillance violates the tenets of a liberal democracy. The troubling nature of this surveillance apparatus is exacerbated by its surreptitious operation, its prevarication, and its resistance to the wishes of a majority of users. Others have eloquently established these claims through systems' analysis, demonstrations and public opinion surveys. [42], [17], [41] Data generated from online surveillance contributes to the creation of valuable, but often highly problematic profiles that fuel the information and behavioral advertising industries with uncertain, potentially negative effects on their subjects. Against this backdrop, we judge the aims of AdNauseam, which include the disruption of this process, to be morally defensible.

The second charge to designers whether obfuscsation impose a lower collateral costs than alternative approaches for achieving similar ends. Comparing the purported cost or damage caused by AdNauseam against alternative approaches involves more measurements and even more uncertainties than we are able to tackle here. But, by the same token, this dearth of concrete evidence also poses a challenge to critics who accuse ad blockers—and would similarly accuse AdNauseam—of harming the web's economy. Even if one holds that the "best" resolution would be societal-level regulation, there has been little progress on this front despite sustained privacy activism. As important as seeking credible alternatives, however, is weighing the purported harms or costs of using AdNauseam.

Among the latter, the harm of "wasting" network bandwidth or server resources is ironic at best, given the vast amount of bandwidth used by advertisers and trackers, the performance degradation resulting from loading this unwanted content, and the financial toll on users paying for fixed data plans. From an ethical perspective, it is questionable whether the term "waste" is appropriate at all. For those who deliberately choose to install and use AdNauseam, it offers utility as a protective shield for privacy and an escape from inappropriate profiling. In our view, these are not worthless endeavors.

One of the most aggressive charges leveled at AdNauseam is that it perpetuates "click fraud." Since obfuscation and fraud both involve forms of lying that disrupt entrenched systems, it is important to evaluate whether the two forms are alike. To carry this out, we consult various definitions: "[Click] fraud occurs when a person, automated script or computer program imitates a legitimate user of a web browser, clicking on such an ad without having actual interest in the target of the ad's link" [30] comes close to capturing AdNauseam in its notion of clicking without actual interest, but this definition seemed overly broad in that it commits users to click only on ads in which they are interested, and seems an unjustifiable restriction on liberty of action. We also argue that if the automated script is performing as an agent of an individual, through that individual's legitimate choice, then the script is a proxy for the user. John Battelle's account [4], which includes motive and intention, gets closer to the standard meaning of "fraud" in "click fraud": the "'decidedly black hat' practice of publishers illegitimately gaming paid search advertising by employing robots or low-wage workers to repeatedly click on each AdSense ad on their sites, thereby generating money to be paid by the advertiser to the publisher and to Google." While elements of the above definitions overlap with AdNauseam's clicking (without genuine interest in their targets), machine automation is only incidental to click fraud, and may instead involve "low-wage workers." More significant is what AdNauseam does not share with click fraud, namely action on behalf of stakeholders resulting in financial gain. In litigated cases of click fraud the intention to inflate earnings has been critical.

We readily admit that a primary aim of AdNauseam is to disrupt business models that support surreptitious surveillance. It does not follow however that AdNauseam is responsible for the demise of free content on the web. First, it is not, as we make clear on the project page, advertising that is the primary target of the project, but rather the tracking of users without their consent. Contextual advertising that does not involve tracking can certainly support free content just as it has in the past. Second, web content is not actually 'free' as this argument implies. The development of the Internet has been supported largely by government funding (and thus by taxpayers) since its beginning. In fact, vast infrastructure and energy costs are still born in large part by taxpayers, not to mention the potentially species-threatening cost to the environment posed by increasing data traffic [24]. Critics may say that adblocking users free ride upon those who allow themselves to be tracked, however, in our view this presumes an entitlement on the part of trackers that is indefensible; one may equally charge trackers with destructive exploitation of users [5]. Lastly, in regard to free riding, we wish to point out that the hiding of ads is an optional element of AdNauseam, one that users must explicitly opt into when they install the software.

IV. Related Work

The strategy of obfuscation has been broadly applied—in search [27], location tracking [33], social networks [31], anonymity [8], [39], etc.—and, as such, has been recognized as an important element of the privacy engineer's toolbox. A range of obfuscation-based projects have been described in [5], including FaceCloak [31], for Facebook profiles, BitTorrent Hydra [39], for decoy torrent sites, and CacheCloak [33], for location data. There have also been a number of obfuscation schemes for web search [3].

Other relevant work, described in [28], has come from the art/tech community. "I Like What I See" is a tool that automatically clicks all 'Like' links on Facebook to obscure user interests. "ScareMail" [19] is an extension built atop Gmail that append an algorithmically-generated narrative containing NSA "trigger-words" to the end of each composed email. "Invisible"[22] extends obfuscation to the conext of genetic privacy via a spray that obfuscates DNA to frustrate identification.

Two early tools addressing surveillance integrate ad-blocking with some broadly-defined social good: AddArt [2] replaces ads with user-configurable art, while AdLiPo [26] does the same with language art. Lightbeam [34], provides displays of users' connections, including advertising networks (though not ads themselves). Floodwatch [14] is the one tool we have found that provide visualizations similar to our own, though it requires communication with a trusted 3rd-party server to do so. Privacy Badger [11] blocks third-party requests, but operates via real-time decisions based on content rather than lists, blocking only those resources engaged in tracking.

V. Future Work

AdNauseam provides individuals with a means expressing a commitment to online privacy without the need to depend on the good will or intervention of third-parties. Although fully functional, AdNauseam is perhaps best considered as a proof of concept for a particular approach to privacy, that is, privacy through obfuscation. As discussed, AdNauseam's potential lies in its capacity to protect individuals against data profiling, as well as simultaneously providing a proactive means of expressing one's views to monolithic and largely uninterested corporations. One key challenge for AdNauseam and similar approaches is a means of providing rigorous, scientific assessments of performance against opaque adversaries; this is to say that we do not (and will not) know precisely the mechanisms that are in place for registering ad clicks, nor precisely the diverse interests of stakeholders in the online advertising ecology.

Going forward, a scientific approach to evaluating AdNauseam's performance, or the performance of any system adopting obfuscation, needs a means of measuring success—namely, evidence that decoy clicks have been registered and have an impact on the resulting profile. Such needs are likely to turn not only on the statistical analysis of signal-to-noise ratios, but also on a practical understanding of how ad-click data is actually mined and used, and the extent to which it influences aspects of user profiles. This would allow future iterations of obfuscation-based tools to be both effective and efficient in the noise they produce.

More concrete future work could take several directions. In the near term we hope to better answer the question of how to perform indistinguishable clicks without leaking user data, as discussed above. Though complex, P2P approaches for the sharing of obfuscation data between users is a potentially ripe area of future work, and might also help address this issue, with users potentially visiting the ads detected by peers as a means of both shielding their data and maximizing indistinguishability. A central challenge here would be meeting functional criteria while not compromising the design constraints discussed early in this paper, e.g., transparency and independence from third-parties.

VI. Conclusions

AdNauseam operates in an environment that is both technologically and socially complex, one in which user data is percieved to be highly valuable. For individuals, however, patterns recorded over time potentially open a window into their lives, interests, and ambitions. Thus surveillance via advertising is not only a source of individual vulnerability, but also interferes with the rights to free and autonomous inquiry, association, and expression that are essential to a healthy democratic society. Consequently, there remain tensions between individual users, collective social and political values, and the economic interests of publishers and advertisers. In a better world, this tension would be resolved in a transparent, trust-based accommodation of respective interests. Instead, concerned users find little transparency and few credible assurances from advertisers that privacy will ever trump the pursuit of profit. Thus trust-based mutual accommodation gives way to an adversarial relationship, one in which we must, as privacy engineers, leverage all the strategies at our disposal. Our success in this endeavor will depend in part on how well we share our experience applying known strategies to new contexts, in concrete and specific detail, according to an evolving set of best practices, as we have attempted above.

We conclude with a philosophical point. In some of the most revealing exchanges we have had with critics, we note a palpable sense of indignation, one that appears to stem from the belief that human users have an *obligation* to remain legible to their systems, a duty to remain trackable. We see things differently; advertisers and service providers are not by default entitled to the externalities of our online activity. Rather, users should control the opacity of their actions, while powerful corporate entities should be held to the highest standards of transparency. Unfortunately this is the opposite of the status quo. The trackers want us to remain machine-readable, so that they can exploit our most human endeavors (sharing, learning, searching, socializing) to extract value and pursue profit. AdNauseam attempts to represent an alternative position.

References

[1] AdBlock Plus. "AdBlock Plus." n.d. https://adblockplus.org/.

[2] AddArt. "AddArt." n.d. http://add-art.org/.

[3] Balsa, Ero, Carmela Troncoso, and Claudia Diaz. "OB-PWS: Obfuscation-Based Private Web Search." *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2012.

[4] Battelle, John. *The Search: How Google and Its Rivals Rewrote the Rules of Business and Transformed Our Culture*. Nicholas Brealey Publishing, 2011.

[5] Brunton, Finn, and Helen Nissenbaum. *Obfuscation: A User's Guide for Privacy and Protest*. MIT Press, 2015.

[6] Cavoukian, Ann, and Michelle Chibba. "Cognitive Cities, Big Data and Citizen Participation: The Essentials of Privacy and Security". *Towards Cognitive Cities*. Springer International Publishing, 2016. 61-82.

[7] Click Fraud. (n.d.). In Wikipedia. Retrieved August 1, 2016. https://en.wikipedia.org/wiki/Click_fraud

[8] Chakravarty, Sambuddho, et al. "Detecting Traffic Snooping in Anonymity Networks Using Decoys." (2011).

[9] "EasyList." 2016. https://easylist.to/

[10] Englehardt, Steven, and Arvind Narayanan. "Online Tracking: A 1-million-site Measurement and Analysis." *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016.

[11] Electronic Frontier Foundation. "Privacy Badger." n.d. https://www.eff.org/privacybadger.

[12] Electronic Frontier Foundation. "Do Not Track." n.d. https://www.eff.org/issues/do-not-track.

[13] Flanagan, Mary, Daniel C. Howe, and Helen Nissenbaum. "Embodying Values in Technology: Theory and Practice." *Information technology and moral philosophy*. (2008): 322-353.

[14] Floodwatch. "FloodWatch." n.d. https://floodwatch.o-c-r.org/.

[15] Friedman, Batya, Daniel C. Howe, and Edward Felten. "Informed Consent in the Mozilla Browser: Implementing Value-Sensitive Design". *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*. IEEE, 2002.

[16] Gervais, Arthur, et al. "Quantifying Web-Search Privacy." *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014.

[17] Goldfarb, Avi, and Catherine Tucker. "Shifts in Privacy Concerns." *The American Economic Review* 102.3 (2012): 349-353.

[18] Gorhill. "uBlock Origin - An efficient blocker for Chromium and Firefox." 2016. https://github.com/gorhill/uBlock

[19] Grosser, Ben. "ScareMail." 2013. Web http://bengrosser.com/projects/scaremail/.

[20] Gürses, Seda, Carmela Troncoso, and Claudia Diaz. "Engineering Privacy by Design." *Computers, Privacy & Data Protection* 14.3 (2011).

[21] Gürses, Seda, Carmela Troncoso, and Claudia Diaz. "Engineering Privacy by Design Reloaded." *Amsterdam Privacy Conference*. 2015.

[22] Dewey-Hagborg, H. "Invisible." 2014. http://www.newmuseumstore.org/browse.cfm/invisible/4,6471.html.

[23] Hansen, Marit, Meiko Jensen, and Martin Rost. "Protection Goals for Privacy Engineering." *Security and Privacy Workshops (SPW)*. IEEE, 2015.

[24] Hazas, Mike, et al. "Are there limits to growth in data traffic?: On time use, data generation and speed." *Proceedings of the Second Workshop on Computing within Limits*. ACM, 2016.

[25] Hoepman, Jaap-Henk. "Privacy Design Strategies." *IFIP International Information Security Conference*. Springer Berlin Heidelberg, 2014.

[26] Howe, Daniel C. "AdLiPo" 2014. http://rednoise.org/adlipo/.

[27] 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.

[28] Howe, Daniel C. "Surveillance Countermeasures: Expressive Privacy via Obfuscation". APRJA, A Peer-Reviewed Journal About Datafied Research 4.1 (2015).

[29] Iachello, Giovanni, and Gregory D. Abowd. "Privacy and Proportionality: Adapting Legal Evaluation Techniques to Inform Design In Ubiquitous Computing." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2005.

[30] Liu, De, Jianqing Chen, and Andrew B. Whinston. "Current Issues in Keyword Auctions". *Business Computing (Handbooks in Information Systems, Vol. 3)* (2009): 69-97.

[31] Luo, Wanying, Qi Xie, and Urs Hengartner. "FaceCloak: An Architecture for User Privacy on Social Networking Sites" *International Conference on Computational Science and Engineering*, 2009.

[32] Mansfield-Devine, Steve. "When advertising turns nasty". *Network Security* 2015.11 (2015): 5-8.

[33] Meyerowitz, Joseph and Romit Roy Choudhury. "Hiding stars with fireworks: Location privacy through camouflage." *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009.

[34] Mozilla. "LightBeam." 2016. https://www.mozilla.org/en-US/lightbeam/.

[35] Murphy, Kate. "The Ad Blocking Wars." ***The New York Times***, 20 Feb. 2016.

[36] Nikiforakis, Nick, et al. "Cookieless monster: Exploring the ecosystem of web-based device fingerprinting." *IEEE symposium on Security and privacy (SP)*. IEEE, 2013.

[37] PageFair, Adobe "The cost of ad blocking–PageFair and Adobe 2015 Ad Blocking Report (2015)."

[38] Regan, Priscilla M. *Legislating privacy: Technology, social values, and public policy.* Univ of North Carolina Press, 1995.

[39] Schulze, Hendrik, and Klaus Mochalski. "Internet Study 2008/2009." *Ipoque Report* 37 (2009): 351-362.

[40] Spiekermann, Sarah, and Lorrie Faith Cranor. "Engineering Privacy." *IEEE Transactions on software engineering* 35.1 (2009): 67-82.

[41] Tucker, Catherine E. "Social networks, personalized advertising, and privacy controls." *Journal of Marketing Research* 51.5 (2014): 546-562.

[42] Turow, Joseph, et al. "Americans reject tailored advertising and three activities that enable it." (2009).

[43] Wills, Craig E., and Doruk C. Uzunoglu. "What Ad Blockers Are (and Are Not) Doing." *Fourth IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*. IEEE, 2016.