Measuring Privacy Loss and the Impact of Privacy Protection in Web Browsing

Balachander Krishnamurthy
AT&T Labs–Research
Florham Park, NJ, USA
bala@research.att.com

Delfina Malandrino*
University of Salerno
Salerno, Italy
delmal@dia.unisa.it

Craig E. Wills
Worcester Polytechnic Institute
Worcester, MA USA
cew@cs.wpi.edu

ABSTRACT

Various bits of information about users accessing Web sites. some of which are private, have been gathered since the inception of the Web. Increasingly the gathering, aggregation, and processing has been outsourced to third parties. The goal of this work is to examine the effectiveness of specific techniques to limit this diffusion of private information to third parties. We also examine the impact of these privacy protection techniques on the usability and quality of the Web pages returned. Using objective measures for privacy protection and page quality we examine their tradeoffs for different privacy protection techniques applied to a collection of popular Web sites as well as a focused set of sites with significant privacy concerns. We study privacy protection both at a browser and at a proxy.

Categories and Subject Descriptors

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

General Terms

Measurement, Performance

Keywords

Web, Privacy

1. INTRODUCTION

The increased dependence on the Internet for a wide variety of daily transactions causes access trails to be left in many locations. There is a corresponding loss in privacy for most users. Virtually all the popular Web sites, either directly or indirectly, gather data about the identity of the users. The growing concerns about identity theft has led users to worry about who has access to information about their Web navigation. Earlier [11], we examined the privacy footprint on the Web as a measure of diffusion of private information. Here, we examine specific techniques to prevent diffusion of private information during Web browsing. Not only do we measure the effectiveness of these techniques, but we also measure the impact of these techniques on the usability of pages when these techniques are applied.

Not all users are equally concerned about privacy loss with concerns varying with the nature of sites visited. However, users should be aware of the nature and extent of private information that is being gathered and measures that could protect their privacy. Most users would like to have control over various bits of private information while still being able to use the Web for their needs.

The key differential arises from the aggregation of private information by third-party sites—sites not directly visited by users and thus potentially opaque to them. Aggregators gather information across various sites visited by a user, tracking sites visited, frequency and duration of visits, primary URLs visited etc.

Private information collected about users can be identifiable bits such as the user's IP address or tailored cookies. Increasingly additional information is being gathered via small scripts sent to the user from third-party sites. Privacy protection thus depends on the nature of private information, who it is being shared with, and what happens when leakage of such information is blocked. On some sites disabling cookies or JavaScript may lead to refusal of service. On others, there may be no visible loss of quality of the pages downloaded. Protection mechanisms must have a clear idea of what is being protected and their impact.

There are two primary goals in this work:

1. Beyond diffusion of private information, we seek to understand what is being diffused and the nature of such diffused information.

2. Examine techniques (both at browser and proxy) for limiting this diffusion and evaluate the trade-offs for these techniques between protecting privacy and impacting the resulting page quality.

We are limited to what we can automatically glean from the set of Web sites visited and it is impossible to track behind-the-scenes data sharing, if any, between sites. We look at a range of privacy-related information and evaluate the effectiveness of various existing and new privacy-protection techniques. We examine techniques applied both at a browser that protect a single user as well as at a proxy

*Work done at AT&T Labs–Research

Symposium On Usable Privacy and Security (SOUPS) 2007, July 18-20, 2007, Pittsburgh, PA, USA..

that protect the privacy of an organization in a more consistent manner and at a potentially lower management cost.

The rest of the paper is divided as follows: Section 2 explores the range of privacy-related information that is gathered and the techniques used to gather it. Section 3 discusses existing and new techniques by which privacy diffusion can be reduced or eliminated, and the impact on the downloaded page as a result of such techniques. Next, Section 4 identifies relevant questions and presents the study we carried out (at browser and proxy) to gather data on the extent of privacy loss and the impact of the various protection techniques. The study results follow in Section 5. Section 6 discusses these results in the context of our study questions. Section 7 presents related work and we conclude with a summary and ongoing work.

2. PRIVACY INFORMATION

In this work we are not interested in authorized private information that a user voluntarily and explicitly provides to a site as part of registering with a site. Rather we are interested in information associated with a user's browsing behavior. A non-exhaustive set of such privacy-related information includes the user's IP address, organization, the `Referer` request field, cookies, search strings, email addresses, passwords, account numbers, etc. Some information such as organization can be inferred from the IP address while other information can be obtained by merging it with other publicly available information.

Among the ways of gathering the private information is to have the user supply it, extract it from the HTTP session (e.g., user's IP address), or send specific scripts (such as JavaScript code) that are executed at the user's browser to enhance a user's browsing experience. However, script execution is potentially a privacy concern as the code can gain access to browser information, such as cached objects and the history of visited links [9]. Cookies are used to identify users, but also to maintain state during transaction sequences. Users may provide information to sites with an expectation that it will not be shared with other sites. But if browsers are asked to access third party sites, these third-party aggregator sites can track user movements via the `Referer` field, cookies, or via "Web bugs," which are small images whose URL contains identifying information about the page being accessed.

3. PRIVACY PROTECTION TECHNIQUES

A variety of techniques applied at a user's browser or at a proxy on behalf of an organization, can be used to protect privacy-related information. Browser-based approaches are attractive because they can be applied at the source of a user's requests and customized to the individual preferences of the user. Proxy-based approaches are independent of the different browsers employed by the users behind it. An anonymizing proxy can be used to hide the source of requests and limit per-user or even per-organization tracking.

There are several techniques already available for privacy protection in modern browsers and proxies; some of them have been introduced to improve page download performance and block ads. These include:

- *Disable cookies.* This technique can be applied to all or just third-party servers. It is the most commonly provided "privacy" technique by browsers such as Internet Explorer (IE) and Firefox. However, the third-party cookie feature is less than ideal on these browsers as one only blocks reads of such cookies and the other only blocks writes [9].

- *Disable JavaScript execution.* Although not labeled as a technique for privacy in either IE or Firefox, this eliminates execution of JavaScript code. The *NoScript* Firefox extension [12] can be used in conjunction with this technique to temporarily or permanently allow JavaScript execution for specific domains.

- *Filter ads.* This technique, provided by the Firefox extension Adblock Plus [1], allows URLs that match regular expressions to be black or whitelisted for download. While not specifically designed for the protection of privacy, this extension can be used to block undesirable URLs. Common filter rulesets (e.g., Filterset.G [6]) do block some servers known to do aggregation.

- *Block images.* This technique is provided by browsers to improve download performance, but because some images are used to communicate information via identifying URLs, the elimination of images provides some privacy protection. Firefox provides a sub-feature allowing only third-party images to be filtered.

We next enumerate other techniques that focus on eliminating content retrieved from third-party servers that can be used to protect privacy:

- *Filter all third-party objects*: Whitelists all content from the domain (2nd-level DNS name) of the user request and filters content from other domains. This technique eliminates all object retrievals that could be used by third-party servers to aggregate information about a user's page retrieval, although this technique may also filter needed content for a page.

- *Remove JavaScript content*: Rather than disable the execution of JavaScript, an alternate technique is to simply remove the content from objects containing it.

- *Filter requests with identifying URLs*: Some URLs are used to pass parameter values and information to the server. These URLs contain characters such as '`?`', '`=`', or '`&`'. Elimination of these URLs reduces the capability of information to be passed as part of the URL, although this filter could remove needed content for the page.

- *Filter objects from top aggregation servers*: Focus on blocking all objects from well-known third-party servers.

- *Remove invisible Web bugs.* Instead of examining the URL for filtering, 0x0 or 1x1 pixel Web bug images can be simply removed.

- *Anonymize cookie content.* Cookies can be provided to a site, but obfuscated to avoid identification or they can be anonymized via a sharing mechanism such as provided by BugMeNot [3] for Web site registration information.

A variety of approaches are thus available to protect the privacy of users. These techniques can be used alone or in conjunction with each other, and applied at the browser or the proxy level. We study many of these techniques and evaluate their effectiveness for privacy protection.

3.1 Impact on Page Functionality and Quality

Privacy protection techniques have the potential to impact the functionality and quality of pages to which they are applied. Users may thus be reluctant to use techniques out of concern that they may "break" their browsing experience or reduce the quality of the retrieved page.

We examine the impact of different techniques on the *normal* browsing experience of the user. We define normal as a browser or proxy configuration that permits *all* object types to be retrieved and has cookies and scripting enabled. Relative to this normal experience, we have identified three types of impact that a privacy prevention technique can have on the browsing experience for a user retrieving a page.

- An *error* occurs: in response to a user's request an explicit error message is shown or the rendered page has no content. The technique employed has caused the page to "break", which may happen for some pages when cookies or JavaScript are disabled.
- A *warning* occurs: a rendered page displays correct content, but a warning message appears as part of the page indicating that functionality may be impaired.
- A *quality degradation* occurs where the page is rendered and no error or warning message appears, but the quality of the rendered page is degraded. Degradation is most apparent when the visible content (text and images) of the page is modified or missing, but it may also result in links or forms that no longer work properly if, for example, they depend on the execution of JavaScript.

We seek to determine objective outcomes of when a page breaks with an error or warning as well as subjective outcomes such as the visible quality of the page being reduced. We need to detect these outcomes in an automated manner that does not require manual intervention. In the next section, we define how these metrics are determined in an automated manner.

4. STUDY

Our study focuses on the following questions concerning privacy loss during Web browsing:

1. What is the nature of privacy information loss when a user visits a Web site? In particular, we are interested in information lost to third-party servers, which are not directly contacted by the user.
2. What privacy protection is afforded by the various privacy protection techniques?
3. What are the tradeoffs between privacy protection and page quality as a result of using these techniques?
4. What is the most appropriate place to implement such techniques—at the browser or at a proxy?

In the remainder of this section we describe the methodology used to answer these questions followed by the results in the next section. Section 6 discusses the results in the context of these questions.

4.1 Test Sets and Basic Testing Methodology

We used two sets of Web sites for this study. The first is chosen to represent a variety of popular sites. It includes a set of English-language sites chosen across various categories from Alexa's popular sites [2], used in earlier work [10]. This set included 100 pages from each of 13 categories: arts, business, computers, games, health, home, news, political, recreation, reference, regional, science, and shopping. With overlaps and multiple pages from the same server, we ended up with pages from over 1000 servers. The second set (used in [11]) focused on Web sites involving the managing of personal fiduciary information. Users provide private information such as credit cards and bank account numbers to such sites. We created nine categories of such sites: credit, financial, insurance, medical, mortgage, shopping, subscription, travel, and utility.

As our basic measuring methodology to gather realistic data about page downloads, we used the Firefox browser augmented by the "Pagestats" JavaScript extension [5]. This extension records information about when each HTTP request was made and the response received in an in-memory table then writes it to a log file when all objects for a page are loaded. The interface allows the extension to run the browser in batch mode where a list of sites is specified. This extension was used to retrieve pages of the "alexa" and "fiduciary" data sets in October/November 2006, using different privacy protection techniques provided at the browser or a proxy.

4.2 First-Party Page Impact

We first examined privacy information and techniques applicable to the first-party servers serving the content directly requested by users. Although these are "authorized" requests, not all users may be comfortable accepting cookies or executing JavaScript even from these first-party servers. We thus initially focus on controls for the use of JavaScript and cookies for first-party servers. Note that the type of access to a site for this test is that of a *casual user* visiting a page of a site rather than a *customer*. A casual user visits a site to browse while a customer visits a site to transact business, which may likely involve registration and logging in to the site.

We used the Pagestats extension to retrieve the contents of all pages in the alexa and fiduciary data sets in three modes: 1) *normal*, with JavaScript and cookies enabled; 2) *nojs*, with JavaScript disabled and cookies enabled; and 3) *nocookie*, with cookies disabled and JavaScript enabled. We then examined the trace of objects requested for the *nojs* and *nocookie* retrievals compared with the normal retrieval. We examined cases where the initially requested URL was directed to a different URL than normal as might occur for redirection to an error page, the number of redirections for the initial URL that was different than normal, and the number of retrieved objects and bytes was significantly smaller than normal (as would be the case for an error or blank page).

We used this methodology to identify pages where disabling JavaScript or cookies for the first-party server causes

the pages to break with an error message. However identifying pages with warning messages is more problematic as the page may render in a manner similar to the normal case. We thus used an alternate approach of downloading page content with the *wget* utility, parsing it to extract text in the HTML `<noscript>` tag, and searching for text such as "script", "browser" or "refresh" indicating a warning message if scripting was disabled.

4.3 Privacy Information Lost to Third-Party Domains

We next examined the nature of privacy information that is lost to third-party servers. Rather than examine individual third-party servers, all servers with the same 2nd-level DNS domain are considered part of the same domain[1].

Note the use of only the domain name to determine whether a server is first-party or third-party is potentially problematic as content providers can use DNS aliases to hide the true name and organization of a server. As part of work in [11], two of the authors examined this issue and uncovered instances where a third-party server did hide behind what appeared to be a first-party domain name. We also found cases where servers appearing to be in different domains were in fact part of the same organization. This analysis was done by examining the authoritative DNS server for each server name. However here we use only the server name itself in making these decisions because the number of such cases is (as yet) relatively small and this approach allows the implementation of the techniques to be simpler.

Beyond the number of third-party domains contacted for each page, we focus on the number of these domains that are from the top aggregation domains. The third-party aggregation domains are the most frequently used ones by pages in our data set and thus in the strongest position to aggregate data. Using our methodology in [11] we found these domains to be `doubleclick.net`, `2mdn.net`, `atdmt.com`, `google-analytics.com`, `2o7.net`, `googlesyndication.com`, `akamai.net`, `advertising.com`, `hitbox.com`, and `questionmarket.com`.

Additionally, we also focus on three specific types of retrievals that have privacy implications in conveying information to third-party servers.

1. The number of unique third-party domains for which at least one object is retrieved and at least one cookie is associated with this domain.

2. The number of unique third-party domains for which at least one object containing JavaScript is retrieved.

3. The number of unique third-party domains for which at least one object with an "identifying" URL is retrieved.

We use data gathered for the two data sets to initially characterize privacy diffusion for different categories and a specific subset of fiduciary sites where there are more concerns of privacy diffusion. These data represent a macro-level potential for privacy diffusion across a large number of sites. The motivation is to measure potential loss and to examine if some categories of sites are more at risk in aiding privacy diffusion.

[1]The servers `a.foo.com` and `b.foo.com` are part of the same `foo.com` domain. If the Top-Level Domain (TLD) is a country code and the TLD is subdivided using recognizable domains such as "com" or "co" then servers are grouped according to the 3rd-level domain.

4.4 Per-User Privacy

Measures for privacy of a category is obviously not a measure of any individual user or organization's privacy diffusion since no user or organization goes to all these sites even when we limit to any particular site category.

One way of obtaining a per-user measure is to make a browser extension available to a large number of users and generate specific measures for each of them. While theoretically feasible, the results would be only as representative as the set of users who can be induced to download the extension and return the results to us. Instead we constructed a small number of differing profiles of users with reasonable settings. We then generated measures for each of them and examined the distribution of the results. The expectation is that many users will find their results to be in the range of results we present.

4.5 Page Quality Measures

Before studying specific privacy protection techniques, we need to define page quality. Earlier in this section, we described techniques to automatically detect when the download of a page "breaks" due to disabling of cookies or JavaScript. However, a more subjective type of impact is when the technique causes a degradation in the quality of the page.

To determine this type of impact we first made informal observations of different pages with different types of techniques applied. For one class of pages and techniques, we observe the technique has minimal impact—possibly removing a few ad images. In other cases, the page is still usable, but missing images. A more severe impact is when the text on the resulting page is still available, but navigation and formatting is affected—possibly due to missing style sheets. In the most severe cases the resulting page is effectively not usable for its intended purpose.

While we can make observations of pages under different circumstances, this approach is both subjective and cannot be applied on the scale needed for this study. Rather, to measure page quality we define a metric that can be deterministically computed based on the objects retrieved for a page when a given technique is applied relative to the objects retrieved under normal conditions. This straightforward metric computes the relative number of visible objects, which we define as HTML and image, as well as the relative number of visible bytes. A metric of 90 and 85 for a given page and technique means that the technique causes 90% of the visible objects and 85% of the visible object bytes to be downloaded for this page relative to a normal retrieval. While these measures may not represent the "true" subjective impact of a technique for a given page, we believe, based on our initial observations, that this is a viable first-order approach to objectively measure the quality impact across a set of pages for a given technique.

4.6 Privacy Protection Techniques

Having defined a means to measure the disclosure of privacy information to third-party servers and to evaluate the quality of a page for a given technique we have the ability to understand their tradeoffs for any privacy protection technique. We apply techniques at both the browser- and

proxy-level. Browser-level techniques are evaluated without the use of a proxy, while proxy-level techniques are evaluated using a browser operating under normal conditions. At both levels the objects retrieved for each page are recorded at the browser using the Pagestats extension. In addition, timing measurements are taken at the proxy to understand the performance impact of the techniques as the proxy could become a bottleneck in handling privacy for a large number of users in an organization. Using the range described in Section 3 we implement a number of techniques.

4.7 Proxy Implementation

Privacy protection at a proxy can provide an added layer of privacy and amortize costs as users behind the proxy do not have to do any work. We used the SISI framework [4], a programmable intermediary infrastructure, built on top of Apache with new Perl modules integrated via the mod_perl mechanism, to implement various protection services. SISI handlers override default handlers in various phases of the Apache Request lifecycle. The *ProxyPerl* handler in the Apache `Pre-Connection` phase, intercepts requests and examines `Referer` field, cookies, OS and browser information, identifying URLs, and blocks requests to third-party servers or redirection to ad links. Services for removing JavaScript code, ads, images and Web bugs are implemented via handlers in the `PerlResponseHandler` phase. We configured the Firefox Pagestats extension to request the pages in the alexa and fiduciary data sets. We route requests through our proxy and measure overhead at request and response handling stages.

Privacy and quality results for the proxy are shown in the following section. The implementation on a RedHat Linux 2.6.9 Dell desktop showed that the processing overhead is minimal—often in the few milliseconds range with most services incurring a overhead of around half a millisecond. Not surprisingly, the services that require parsing of the HTTP response to remove ads and scripts took longer than simply removing a header in the request. Providing *all* the services on request and response took an average of 27 ms for fiduciary sites and 54 ms for news sites (due to the higher content in pages in the latter).

5. RESULTS

We now present the results of our study and what can be inferred from them.

5.1 First-Party Page Impact

The first portion of our study examined the impact of disabling all cookies and JavaScript when downloading the contents of pages in our study set. Servers set cookies for 187 (16%) of the 1123 pages in the alexa study set for which pages were successfully retrieved. Sites test for the acceptance of cookies by sending redirects on initial access and then testing whether cookies are returned by the client when it follows the redirection. If cookies are disabled then only 5 (<0.5% of the 1123) of these pages break with messages indicating that the site cannot be visited without enabling cookies for the first-party server. These were manually validated and include well-known sites such as `www.expedia.com`, `www.gap.com` and `www.netflix.com`.

We find that servers set cookies for 52 (67% of the 78) fiduciary data set pages and 3 (4% of the 78) of these pages break with an error message if cookies are disabled. Cookies are used much more frequently by the fiduciary pages. The large gap between the number of first-party servers that set cookies and the number that require cookies just to visit the site indicates cookies are often not needed for the casual visitor. On the other hand, customers of a site may be required to enable cookies. Manual exploration of sites of the fiduciary pages indicates that many of them do require cookies to be enabled to register or login to the site.

Looking at the impact of disabling JavaScript, we find more cases where the page breaks. In the alexa data set, 30 (<3%) pages result in an error when JavaScript is disabled. Two types of results are classified as errors. In one, an explicit error message is shown, similar to what is shown for disabled cookies, indicating that JavaScript is required. The second type of error occurs more frequently where the rendered window is blank because the page content has been constructed to depend on the interpretation of the embedded JavaScript.

In addition to errors, disabling JavaScript also causes warning messages to appear in the rendered content for another 45 (4%) of alexa pages. These messages were discovered by searching the content contained within the `<noscript>` tag for words such as "script" or "browser" indicating an error. The messages indicate that features of the page or site will not work properly because JavaScript is disabled. Looking at the fiduciary data set we find that 5 (6%) of the pages result in an error with JavaScript disabled and another 11 (14%) result in a warning message. The need for JavaScript is greater across the smaller fiduciary data set than the larger, and more diverse, pages of the alexa data set.

Overall, the impact of disabling JavaScript for content retrieved from first-party servers is more problematic than disabling cookies. This impact is also greater because disabling JavaScript may not result in an explicit message that it is needed, but rather the page may not render without any indication of the problem. In addition, disabling JavaScript from first-party servers introduces subtle problems where links or forms may fail due to dependence on JavaScript execution.

As an important follow-on to these results, we examined what happens to the pages that break when first-party cookies or JavaScript are enabled, but third-party cookies or JavaScript are disabled. In these cases, the pages no longer break in terms of showing error messages or a blank window. This outcome means the privacy protection techniques applied to third-party servers may impact the quality of the resulting pages, but do not break the pages.

5.2 Privacy Information Lost to Third-Party Domains

We next examined the the number of third-party domains contacted for each page retrieved and focused on four specific privacy implications arising from these third-party domain contacts. Table 1 shows average per-page results for each of the categories in the alexa data set. The results show that on average 2.9 third-party domains are accessed for all pages in the data set, of which 1.2 (41%) are in the top-10 domain set. Next we examine the specific privacy implications of these accesses. Of the 2.9 domains, 1.4 of them have at least one cookie associated with the access (0.7 or 50%, from a top-10 domain), 2.1 retrieve at least one object with an identifying URL (1.0 or 47%, from a top-

10 domain), and 1.1 retrieve at least one JavaScript object (0.5 or 45% from a top-10 domain). The results show that on average pages in the news and political categories access the most third-party domains while pages in the health and science categories access the least number of these domains. The top-10 domains constitute a significant fraction of the third-party domains in all cases.

Table 1: Average Number of Third-Party Domains Accessed and Privacy Implications per Page by Categories of Alexa Data set

	3rd-Party Domains	Privacy Implications		
		cookie	ident	js
Category	all/top	all/top	all/top	all/top
all	2.9/1.2	1.4/0.7	2.1/1.0	1.1/0.5
arts	4.1/1.8	2.1/1.1	3.0/1.5	1.6/0.8
business	3.2/1.4	1.9/1.1	2.4/1.1	0.9/0.4
computers	2.4/0.9	1.2/0.6	1.6/0.7	0.9/0.3
games	2.7/1.1	1.4/0.7	2.1/1.0	1.0/0.5
health	1.8/0.9	0.7/0.4	1.4/0.7	0.9/0.5
home	3.7/1.9	1.7/1.0	2.8/1.6	1.6/0.9
news	5.5/2.1	2.6/1.4	3.9/1.8	2.2/0.8
political	4.5/1.3	1.9/0.6	3.1/1.1	1.7/0.7
recreation	2.1/1.1	1.2/0.7	1.6/0.8	0.6/0.3
reference	1.5/0.7	0.7/0.4	1.2/0.6	0.7/0.4
regional	2.4/1.0	1.2/0.7	1.7/0.8	0.9/0.3
science	1.5/0.7	0.7/0.4	1.0/0.6	0.6/0.3
shopping	2.9/1.0	1.6/0.6	1.8/0.6	0.5/0.2

The specific privacy implications that we focus on are mutually exclusive and the retrieval of objects from a particular domain could involve any number of these implications. For example, the retrieval of an image with a non-identifying URL from a non-top-10 third-party server with no cookie set has none of these specific privacy implications. This result implies that the third-party access raises a relatively low concern from the standpoint of privacy. On the other hand, consider the retrieval of a third-party object found on pages in the alexa data set with the URL `http://www.google-analytics.com/urchin.js`. Once downloaded, this script causes a subsequent 514-character identifying URL for a 35-byte image object `http://www.google-analytics.com/__utm.gif?utmwv=1&...` to be downloaded. No cookies are set by the server. Retrievals from this domain have two of our specific privacy implications as they include a JavaScript object retrieval and an identifying URL. In addition, this interaction is with a top-10 third-party domain. Such third-party accesses with multiple privacy implications, particularly with a top-10 domain, are problematic from the standpoint of privacy protection.

Table 2 shows similar per-category results for the fiduciary data set. The results show that overall the fiduciary pages have fewer privacy concerns, but that categories such as subscription have as many privacy implications as categories in the alexa data set.

5.3 Per-User Privacy

While the previous results provide privacy implications for categories of pages, they do not provide privacy implications for the set of sites that a particular user visits. We have chosen to illustrate per-user results by selecting random "bags" of pages across the various categories that reflect particular user profiles. This approach allows us to understand the range of privacy implication results that can be expected for individual users.

Table 2: Average Number of Third-Party Domains Accessed and Privacy Implications per Page by Categories of Fiduciary Data set

	3rd-Party Domains	Privacy Implications		
		cookie	ident	js
Category	all/top	all/top	all/top	all/top
all	1.8/0.7	1.3/0.5	1.3/0.5	0.5/0.2
credit	1.8/0.7	1.2/0.7	1.2/0.7	0.3/0.2
financial	1.1/0.4	1.1/0.4	0.8/0.2	0.3/0.0
insurance	0.6/0.1	0.1/0.1	0.2/0.1	0.1/0.0
medical	0.2/0.1	0.1/0.1	0.0/0.0	0.0/0.0
mortgage	0.8/0.3	0.5/0.2	0.7/0.2	0.2/0.0
shopping	3.1/1.3	2.4/1.0	2.4/1.0	0.6/0.2
subscription	6.0/2.2	3.9/1.4	4.6/1.8	2.3/0.8
travel	2.2/1.3	1.7/0.9	1.3/0.8	0.4/0.3
utility	0.5/0.0	0.5/0.0	0.5/0.0	0.1/0.0

First looking at a profile representing a user with equal interests from all categories in the alexa data set, we find profiles with a range of 0.7–3.7 third-party domains accessed. This is a broad range around the 2.9 average shown in Table 1. Similarly if we look at a user profile with a single page drawn from each of the fiduciary categories then we obtain 0.3–3.4 third-party domains accessed, which is a broad range around the 1.8 average shown in Table 2. Ranges for specific privacy implications in Table 2 are not as large, but the upper-bound of each is about twice the average shown in the table.

Finally, we examined a more realistic profile from our fiduciary set with more expected pages in categories such as financial, shopping and subscription and fewer in medical and utility. This profile exhibits similar upper-bounds as the previous profile with larger lower-bounds for each the measures.

Table 3: Privacy Prevention Techniques Explanation

norm	normal browsing behavior
nocookie	disable all cookie
no3cookie	disable all third-party cookies
nojs	disable all JavaScript execution
no3js	disable all third-party JavaScript execution
noimg	filter all images
no3img	filter all third-party images
notop	filter all objects from top-10 3rd-party domain
noad	filter all ad object
no3obj	filter all third-party objects.
no3objnoid	filter third-party objects but allow those with non-identifying URLs
nowebbug	filter out "Web bug" images
noidheader	filter out potentially identifying headers in the request

Table 4: Privacy Prevention Techniques and Their Implementation

Technique	Browser Implementation	Proxy Implementation
norm	Browser features enabled.	Proxy does not filter requests or responses.
nocookie	Use browser feature to disable cookies.	Proxy removes all `Set-Cookie:` headers in the response.
no3cookie	Use browser feature to disable third-party cookies.	Proxy removes all `Set-Cookie:` headers in the response for third-party object requests.
nojs	Use browser feature to disable JavaScript execution.	Proxy filters out all code between the HTML tags `<script>` and `</script>` in HTML content.
no3js	Use NoScript extension with a list of allowed domains.	Not implemented.
noimg	Use browser feature to loading of images.	Proxy filters the HTML response content to remove all `<IMG>` tags.
no3img	Use browser feature to loading of images only for originating site.	Not implemented.
notop	Use the Adblock Plus extension to blacklist top-10 third-party domains listed in Section 4.	Not implemented.
noad	Use the Adblock Plus extension with Filterset.G rules to blacklist all objects matched as ads.	Proxy returns an HTTP Not Found message to the browser for all objects matching a rule in the Filterset.G ruleset.
no3obj	Use the Adblock Plus extension with rules to blacklist all objects then whitelist objects from domains in the retrieval set.	Proxy returns an HTTP Not Found message to the browser for all objects where the server portion of the URL does not match the domain of the base URL for the page.
no3objnoid	Use the Adblock Plus extension with rules to blacklist all objects then whitelist objects from domains in the retrieval set. Also whitelist URLs not containing a '`?`', '`=`', or '`&`'.	Proxy returns an HTTP Not Found message to the browser for objects where the server portion of the URL does not match the domain of the base URL for the page, except the proxy allows requests not containing a '`?`', '`=`', or '`&`'.
nowebbug	Not implemented.	Proxy filters the HTML response content to remove all `<IMG>` tags where the displayed size of these images is 0x0 or 1x1 pixels.
noidheader	Not implemented.	Proxy filters out the request headers `User-Agent`, `From`, `Referer`, and `Cookie`.

5.4 Privacy Protection Techniques

Given that we defined and measured specific privacy implications when third-party domains are contacted during the retrieval of a page, we focus on the effectiveness of different techniques to protect privacy. A range of possible techniques was outlined in Section 3. Table 3 defines a short-hand notation for each technique that is used in subsequent tables and graphs. Table 4 describes how each of these techniques is implemented at the browser and proxy. The techniques deal with handling cookies and JavaScript code, how different types of objects are handled, and which domain serves the object. For some techniques the browser and proxy implementation is pretty much the same, while in other cases different approaches were taken to implement the same technique. In a few cases, a technique was not implemented at either the browser or the proxy. These cases occurred when it was easier to implement a technique with one entity than the other.

We begin our study of these techniques by examining their impact on the quality of the downloaded page. As indicated in Section 4 we focus on the visual content by examining the number of objects and bytes that are retrieved for a page. Using this definition, Table 5 shows the quality results

Table 5: Page Quality for Browser Techniques of News Pages Using Object and Byte Percentage

	Page Percentiles (using Object %)				Ave. Object	Ave. Byte
Technique	1-25	26-50	51-75	76-100	%	%
norm	0	0	0	100	100	100
nocookie	2	0	0	98	96	93
no3cookie	1	0	1	98	97	94
nojs	4	7	28	61	77	77
no3js	0	1	11	88	90	87
noimg	90	5	1	4	15	26
no3img	12	0	10	77	80	80
notop	1	1	2	96	92	90
noad	0	3	21	76	84	80
no3obj	14	2	16	67	75	74
no3objnoid	2	1	10	86	89	85

Table 6: Comparison of Privacy Protection Technique Effectiveness Versus Page Quality for News Pages with Browser Implementation

Technique	3rd-Party Domains all/top	Privacy Implications cookie all/top	ident all/top	js all/top	Cum. Priv. Implications all/top	Object Quality %	Byte Quality %
norm	5.5/2.1	3.3/1.3	4.1/1.7	2.1/0.8	9.5/3.8	100	100
nocookie	5.5/2.0	0.0/0.0	3.8/1.6	2.1/0.7	5.9/2.3	96	93
no3cookie	5.8/2.3	0.0/0.0	4.2/1.9	2.3/0.9	6.5/2.8	96	93
nojs	2.0/0.8	1.1/0.5	0.8/0.4	0.0/0.0	1.9/0.9	77	77
no3js	2.5/0.9	1.3/0.7	1.3/0.6	0.0/0.0	2.6/1.3	90	87
noimg	3.6/1.5	2.1/0.9	2.8/1.2	2.1/0.8	7.0/2.9	15	26
no3img	3.4/1.2	2.0/0.7	2.5/1.0	2.0/0.7	6.5/2.4	80	80
notop	3.2/0.0	1.5/0.0	1.9/0.0	1.3/0.0	4.7/0.0	92	90
noad	1.9/0.1	0.6/0.0	0.9/0.1	0.7/0.1	2.2/0.2	84	80
no3obj	0.3/0.0	0.1/0.0	0.1/0.0	0.1/0.0	0.3/0.0	71	69
no3objnoid	2.3/0.3	0.8/0.1	0.1/0.0	1.3/0.2	2.2/0.3	86	84

for each of the techniques implemented at the browser for the pages in the news category of the alexa data set—the category in Table 1 with the most privacy implications. We focus on one category to control the number of pages to be tested for each technique.

The basic calculation done for each page in the set is to determine the number of visible objects retrieved under normal conditions and the number of objects retrieved for a given technique. Dividing the latter by the former and converting to a percentage is our object quality for that page. If it happens that the number of objects retrieved for the technique is more than the normal case then the quality is assigned to be 100%. This case, which does not happen often, may mean that more objects are actually retrieved with the technique or the contents of the page have changed because the techniques were tested at a similar, but not the same time. In a similar manner, the byte quality is based on the number of visible bytes retrieved for the normal case and with the technique are compared. The average of the object and byte quality for all pages in the news category are shown in the last two columns of Table 5. The previous four columns in the table show the percentage of pages in each of four percentile ranges for object quality. The distribution for byte quality in percentile ranges is similar.

In examining the results, disabling cookies causes only 2% of the pages to have the number of objects drop to 1-25% of normal conditions, but turning off the download of images causes 90% of the pages to have an object drop in this range. The overall results show that the *noimg* technique not surprisingly has the most severe impact on page quality, but that the filtering out all third-party objects or images has a non-trivial impact. In examining the impact of the techniques on privacy, we use the average object and byte percentages as measures of page quality.

Table 6 shows the impact of the different browser-based implementations of the privacy protection techniques for the news category pages. On average for each technique, it shows the number of third-party domains contacted for each page and those in the top-10, the number for each of the privacy implications for each page and those in the top-10, the cumulative privacy implications (for all and top-10 third-party servers), and the object and byte measures of quality. We include the cumulative privacy implications values as quick comparison across techniques where the lower the value the better. Potential users of such techniques should examine the privacy implications that are of most concern to them.

In examining the results, the best technique is one that has the most impact on reducing privacy implications while having the least impact on the page quality. By that criteria the *noimg* technique is the worst as it has a large impact on page quality while not showing a significant improvement for privacy concerns. The *no3obj* technique provides the best privacy results because it prevents the retrieval of all third-party objects. Note we see non-zero values for the privacy implications because of implementation limitations for this technique with Adblock Plus. Currently this technique is implemented by whitelisting all first-party domains in the data set and in a few cases pages in one first-party domain retrieve objects from another first-party domain. Aside from this minor implementation issue, the real problem with the *no3obj* technique is that it provides the worst quality of any technique other than *noimg* because enough pages use third-party content distribution domains to serve objects.

To better illustrate the tradeoffs between privacy protection and page quality for these techniques, Figure 1 shows the relative performance of each technique using the object percentage in Table 6 as the measure of page quality and the cumulative privacy implications value for all third-party servers as the measure of privacy protection. By definition, the *norm* technique is in the upper-right corner of the graph and an ideal policy would be represented in the lower-right corner. The *no3objnoid* technique, which filters out all requests to third-party servers except those that do not contain an identifying URL, provides more reasonable tradeoffs between quality and privacy than the *no3obj* technique, although the *nojs*, *noad*, and *no3js* techniques provides similar tradeoffs in Figure 1.

Figure 2 shows similar tradeoffs, but focuses on privacy implications from the top-10 third-party servers. Not surprisingly, the *notop* technique provides the best privacy/protection tradeoff, but again the *noad* and *no3objnoid* techniques provide close to the same tradeoffs. Results from both Figures 1 and 2 indicate that the *no3objnoid* technique provides reasonable tradeoffs for the privacy implications we have defined, but it could be improved by adding some control on

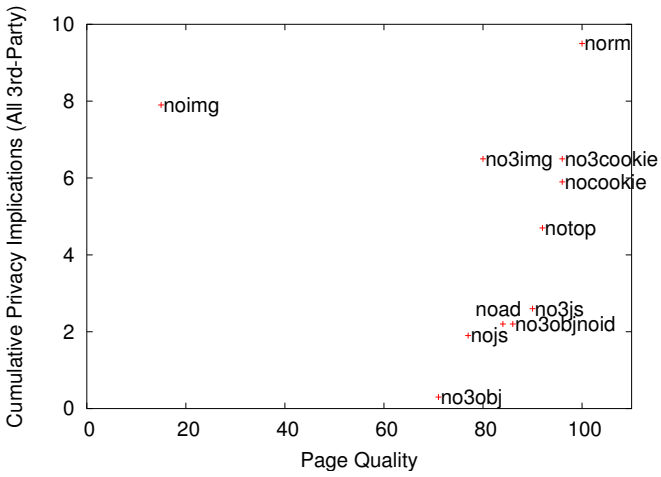

Figure 1: Cumulative Privacy Implications vs. Quality for News Pages with Browser Implementation

third-party cookies or JavaScript along with controlling access to top third-party servers. The *noad* technique also provides reasonable tradeoffs, but requires that an explicit set of filtering rules be maintained over time.

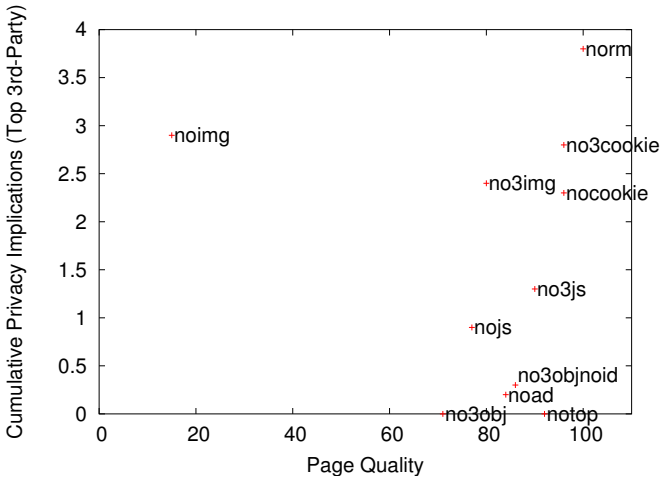

Figure 2: Cumulative Privacy Implications from Top Third-Party Servers vs. Quality for News Pages with Browser Implementation

Table 7 shows similar privacy protection versus quality tradeoffs for the same techniques using the pages in the fiduciary data set. The privacy implications for these pages is on average much less than for the set of news pages, but the relative effectiveness of the techniques is similar.

Table 8 shows the same set of results for the techniques that were implemented at the proxy. The proxy results were recorded by the PageStats extension at the browser exactly the same as for the browser-based techniques, but the techniques themselves were implemented at the proxy. Since *https* requests set up a tunnel between the client and the origin server and the proxy acts as a blind relay we did not modify such requests and pages using this protocol (two pages in the news set and ten in the fiduciary set were dropped for these tests). Cumulative privacy implications for all third-party servers versus quality are compared for these data in Figure 3.

The table and figure results for the various techniques are generally comparable to those implemented at the browser, particularly given that although the same set of pages were tested, the timeframe of the two tests was different. One notable result between the browser and proxy implementations is the low byte quality for the proxy implementation of the *nojs* policy. This difference in not surprising because the proxy implementation removes script code from the content while the browser implementation simply does not execute it. Results for the proxy techniques applied to the fiduciary data set are similar in tone.

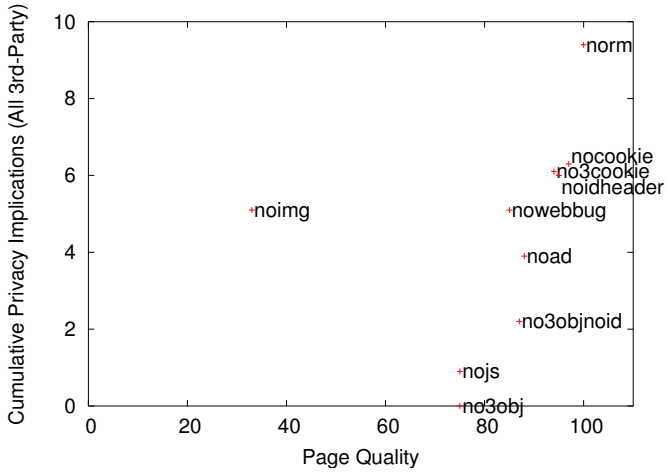

Figure 3: Cumulative Privacy Implications vs. Quality for News Pages with Proxy Implementation

6. DISCUSSION

In Section 4 we raised a number of questions concerning the loss of privacy information during Web browsing. We now examine the results of this study in the context of these questions.

6.1 Nature of Privacy Loss

We focused on the nature of privacy loss to third-party servers because these are the servers visited indirectly by user action. In particular we focused on those accesses that provide opportunity for third-party servers to uniquely identify the actions of users across various first-party sites. These accesses include those using cookies, identifying URLs and JavaScript. We also examined behavior of accesses to the top third-party servers as these servers are in a position to aggregate information across a large number of first-party sites.

We found that all of the three privacy implications are present across our data sets with identifying URLs, then cookies, then JavaScript in terms of use of the largest data set. Roughly 50% of these implications come from the top-10 third-party servers, which are also more likely to use these techniques than other third-party servers.

Table 7: Comparison of Privacy Protection Technique Effectiveness Versus Page Quality for Fiduciary Data set with Browser Implementation

Technique	3rd-Party Domains all/top	Privacy Implications cookie all/top	ident all/top	js all/top	Cum. Priv. Implications all/top	Object Quality %	Byte Quality %
norm	1.9/0.8	1.3/0.6	1.3/0.5	0.5/0.2	3.1/1.3	100	100
nocookie	1.6/0.5	0.0/0.0	1.1/0.4	0.4/0.1	1.5/0.5	95	93
no3cookie	2.0/0.8	0.0/0.0	1.4/0.5	0.5/0.2	1.9/0.7	97	95
nojs	0.9/0.4	0.7/0.3	0.4/0.2	0.0/0.0	1.1/0.5	76	75
no3js	1.3/0.6	0.9/0.5	0.7/0.3	0.0/0.0	1.6/0.8	93	91
noimg	0.9/0.3	0.6/0.2	0.6/0.2	0.4/0.1	1.6/0.5	10	19
no3img	1.0/0.4	0.6/0.2	0.7/0.3	0.5/0.2	1.8/0.7	86	83
notop	1.0/0.0	0.7/0.0	0.7/0.0	0.3/0.0	1.7/0.0	95	94
noad	0.6/0.0	0.3/0.0	0.3/0.0	0.2/0.0	0.8/0.0	90	86
no3obj	0.1/0.0	0.1/0.0	0.1/0.0	0.0/0.0	0.2/0.0	80	80
no3objnoid	0.8/0.3	0.5/0.2	0.1/0.0	0.2/0.0	0.8/0.2	92	90

Table 8: Comparison of Privacy Protection Technique Effectiveness Versus Page Quality for News Pages with Proxy Implementation

Technique	3rd-Party Domains all/top	Privacy Implications cookie all/top	ident all/top	js all/top	Cum. Priv. Implications all/top	Object Quality %	Byte Quality %
norm	5.6/2.2	3.2/1.4	4.1/1.8	2.1/0.7	9.4/3.9	100	100
nocookie	5.8/2.3	0.0/0.0	4.2/1.9	2.1/0.8	6.3/2.7	97	94
no3cookie	5.5/2.0	0.0/0.0	4.0/1.7	2.1/0.7	6.1/2.4	94	93
nojs	1.2/0.2	0.5/0.2	0.4/0.1	0.0/0.0	0.9/0.3	75	24
noimg	2.8/1.1	1.6/0.6	2.0/0.8	1.5/0.5	5.1/1.9	33	26
noad	2.7/0.3	1.2/0.2	1.6/0.2	1.1/0.1	3.9/0.5	88	87
no3obj	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	74	76
no3objnoid	2.2/0.3	0.9/0.1	0.1/0.0	1.2/0.2	2.2/0.3	87	89
nowebbug	3.5/1.1	1.7/0.6	2.0/0.8	1.4/0.4	5.1/1.8	85	27
noidheader	5.5/2.1	0.0/0.0	4.0/1.7	2.0/0.7	6.0/2.4	95	93

6.2 Techniques for Privacy Protection

We examined a wide-ranging set of techniques for protecting privacy—some of these techniques focused on a particular privacy concern, such as cookies or the use of JavaScript, while other techniques, such as elimination of ads or images, were less directly focused on privacy.

Not surprisingly, we found a range of effectiveness for these techniques given the privacy concerns of our study. A commonly available technique to restrict the use of cookies does eliminate that privacy concern, but does not control other loss of privacy. Disabling JavaScript execution better controls privacy loss, but can impact page quality. Elimination of accesses to all third-party servers is the most effective privacy prevention technique, but can have more page quality impact. Given concerns about privacy loss to the top third-party servers, a technique to prevent accesses to these servers is effective from that perspective.

6.3 Tradeoffs Between Privacy Protection and Page Quality

The central question of this work is not just the effectiveness of these privacy protection techniques, but their impact on the user browsing experience. We first examined the impact of two commonly available techniques—disabling cookies and JavaScript—on "breaking" pages. When we deployed these techniques for first-party content in our larger data set we found only a few first-party sites that returned an error for a casual visit if cookies were not enabled. But up to 7% of pages showed a warning or error message or simply did not display if JavaScript was disabled. We found that 20% of pages show a warning or error message for a casual visit to a set of fiduciary-related sites. These results indicate that disabling JavaScript for first-party sites is problematic for breakage. However, disabling JavaScript from third-party sites does *not* break the corresponding pages.

We next examined the tradeoffs between privacy protection and page quality for the range of techniques. Not surprisingly, preventing the download of images provides incomplete privacy protection and has a negative quality impact. Disabling cookies generally has no page quality impact, but provides incomplete privacy protection. Prevention of all third-party object accesses is the best for privacy protection, but other than eliminating images causes the most quality impact. A less restrictive technique that eliminates only third-party accesses with identifying URLs provides reasonable tradeoffs between privacy protection and page quality. We believe that a combination of techniques using this technique as a basis with additional controls on JavaScript, cookies, and known aggregations servers would provide the most effective technique.

6.4 Privacy Protection at the Browser vs. the Proxy

Our study also allowed us to examine the implementation of privacy protection at the browser or at a proxy. From the standpoint of the specific techniques, the two entities were comparable in the range of techniques that could be implemented and their effectiveness. The extension-based approach does allow tighter coordination with the functioning of the browser. One example is the ability to disable JavaScript execution in the browser versus removing the JavaScript code in the proxy.

In terms of the relative merits for wider deployment in an organization, the tradeoffs mimic decisions for other issues such as spam filtering. The browser-based approach allows control according to individual preferences, while the proxy-based approach is browser-independent, requires no user involvement and ensures a degree of privacy protection across the entire organization. However, the proxy-based approach is more difficult to modify if unwanted page impacts for a user occur. It also requires all requests to be routed through it, but our measurement of the proxy overhead found it to be on the order of a few milliseconds for most techniques.

7. RELATED WORK

Although there is considerable prior work on privacy, we are not aware of specific work that attempts to either measure privacy on this scale or evaluates privacy protection.

The Onion Router (TOR [16]) is capable of supplying a series of different IP addresses to those who access it. The Torpark [17] browser is a variant of Firefox that can be run from any public terminal, such as those found in cyber-cafes (directly from a USB drive). Torpark sets up an encrypted connection to TOR and enables a varying set of IP addresses to be sent to a Web site. A proxy could thus have a permanent connection to TOR and route all requests for its clients and provide stronger privacy protection to all its clients. The browser session is slower than with an unmodified browser and the recommendation is to run it with images turned off. The delay is presumably due to the fact that Tor can route a connection through multiple Tor servers (additional level of privacy by distributing privacy). There are over seven hundred Tor servers available already. Note however, that JavaScript executed at a browser can reveal the IP address of the browser!

SecretSurfer [15] is a commercial product capable of routing a session through different proxies at random with automatic storage of passwords and form content, cleaning up of local information left behind by users (cache, download history, search strings etc.) and a JavaScript ad-filter.

Privoxy [14] is a privacy enhancing proxy with filtering capabilities to protect privacy. It provides some of the same capabilities tested with our proxy by filtering Web page content, managing cookies and removing ads.

Greenborder [7], a commercial product, attempts to wall off the user in a virtual environment similar to FreeBSD's jail. Although the primary goal of the product is to provide security in the form of protection against viruses and malware, avoiding DNS spoofing etc., the tool limits personal information about the user by the concept of a privacy zone to not last beyond a session. Additionally, Greenborder incorporates a concept of trusted Web content by limiting Web servers that belong to a range of IP addresses. Earlier work used a similar approach of executing remotely obtained code in a protected environment [13, 8].

Jackson, et al [9] developed two Firefox extensions, SafeCache and SafeHistory, that the apply the same-origin principle to all cache and history accesses. This principle states that only the site that has stored cache or history information in the browser is allowed to subsequently access it. These extensions help to prevent unauthorized access to browser state by script code.

8. SUMMARY AND FUTURE WORK

In this work we have studied the disclosure of privacy information to third-party servers and have a made a significant contribution in evaluation of new and existing techniques for limiting this disclosure. Our work is novel in that we not only study the effectiveness of these techniques for privacy protection, but we also evaluate the techniques for their impact on page quality. This bimodal evaluation approach allows us to understand the tradeoffs incumbent with these types of techniques. In addition, we study techniques implemented at both the browser and proxy, which affords us two vantage points for how techniques can best be implemented.

Our results show that the disabling first-party cookies breaks a relatively tiny set of sites while the disabling third-party cookies provides some privacy protection with minimal page quality loss. More significant privacy protection is provided by techniques that disable or filter out JavaScript content, but these techniques have a more negative impact on pages breaking and page quality. The best technique for privacy protection is to filter out all third-party object retrievals, but this technique has an even more negative impact on page quality. The best techniques we evaluated, in terms of privacy vs. quality tradeoffs, are ones that allow third-party objects to be selectively retrieved. These techniques avoid ads, top aggregation servers and objects with identifying URLs. We were able to implement these techniques at both a browser and proxy. Performance results at the proxy indicate a negligible overhead.

In terms of directions for future work, thus far we have generally studied "single-purpose" techniques that each focus on a single aspect of protection. This approach has allowed us to study the merits of each technique in isolation, but the best approach is likely to be one that is a combination of techniques.

Another direction of future work is to develop a tool, such as a Firefox extension for a user or a proxy for an organization, that can be used to control privacy protection. This tool would provide users a single medium to set privacy protection and allow users to better understand the page quality tradeoffs on the Web sites they access. Related to this direction is ongoing work to extend tools such as AdBlock and NoScript to have settings for automatic control of third-party content instead of requiring explicit filtering rules for each new site accessed by a user.

9. REFERENCES

[1] Adblock plus. http://adblockplus.org/.

[2] Alexa: Most popular web sites. http://www.alexa.com/.

[3] bugmenot.com, tell everyone you know. http://www.bugmenot.com/.

[4] M. Colajanni, R. Grieco, D. Malandrino, F. Mazzoni, and V. Scarano. A scalable framework for the support of advanced edge services. In *Proc. of the International Conference on High Performance Computing and Communications*, pages 1033–1042, Sept 2005.

[5] S. DeDeo. Pagestats, May 2006. http://www.cs.wpi.edu/~cew/pagestats/.

[6] Official home of Filterset.G. http://www.pierceive.com/.

[7] Greenborder keeps you safe on the internet. http://www.greenborder.com/.

[8] S. Ioannidis, S. Bellovin, and J. Smith. Sub-operating systems: A new approach to application security. In *Proceedings of the 10th ACM SIGOPS European Workshop*, Saint-Emilion, France, Sept. 2002. ACM.

[9] C. Jackson, A. Bortz, D. Boneh, and J. C. Mitchell. Protecting browser state from web privacy attacks. In *Proceedings of the International World Wide Web Conference*, Edinburgh, Scotland, May 2006.

[10] B. Krishnamurthy and C. Wills. Cat and mouse: Content delivery tradeoffs in web access. In *Proceedings of the International World Wide Web Conference*, Edinburgh, Scotland, May 2006.

[11] B. Krishnamurthy and C. E. Wills. Generating a privacy footprint on the Internet. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, Rio de Janeiro, Brazil, October 2006.

[12] Noscript. https://addons.mozilla.org/firefox/722/.

[13] V. Prevelakis and D. Spinellis. Sandboxing applications. In *Proceedings of the USENIX Conference FREENIX Track*, Boston, MA USA, June 2001.

[14] Privoxy. http://www.privoxy.org/.

[15] Secretsurfer. http://www.gxsoftware.net/secretsurfer/.

[16] Tor: Anonymity online. http://www.tor.eff.org.

[17] Torpark: Turn any internet terminal into a secure connection. http://www.torrify.com.nyud.net:8080/.