A Survey of the Use of Adobe Flash Local Shared Objects to Respawn HTTP Cookies

Aleecia M. McDonald and Lorrie Faith Cranor

January 31, 2011

CMU-CyLab-11-001

CyLab
Carnegie Mellon University
Pittsburgh, PA 15213

A Survey of the Use of Adobe Flash Local Shared Objects to Respawn HTTP Cookies

Aleecia M. McDonald and Lorrie Faith Cranor*
Carnegie Mellon University

January 31, 2011

Abstract

Website developers can use Adobe's Flash Player product to store information locally on users' disks with Local Shared Objects (LSOs). LSOs can be used to store state information and user identifiers, and thus can be used for similar purposes as HTTP cookies. In a paper by Soltani et al, researchers documented at least four instances of "respawning," where users deleted their HTTP cookies only to have the HTTP cookies re-created based on LSO data. In addition, the Soltani team found half of the 100 most popular websites used Flash technologies to store information about users. Both respawning and using LSOs to store data about users can reduce online privacy. One year later, we visited popular websites plus 500 randomly-selected websites to determine if respawning still occurs.

We found no instances at all of respawning in a randomly-selected group of 500 websites. We found two instances of respawning in the most popular 100 websites. While our methods are different from the Soltani team and we cannot compare directly, our results suggest respawning is not increasing, and may be waning. As in the Soltani study, we found LSOs with unique identifiers. In the 100 most popular websites, LSOs were set at 20, and 9 used their LSOs to store unique identifiers. In 500 randomly selected sites, LSOs were set at 41, and 17 used their LSOs to store unique identifiers. Unique identifiers may, or may not, be keys into back-end databases to perform cookie-style tracking. However, unique identifiers could be benign, for example, uniquely identifying a specific animation or music clip. While we can use contextual information like variable names to guess what a given unique identifier is for, using our study methods we cannot conclusively determine how companies use unique identifiers. We cannot quantify how many, if any, sites are using unique identifiers in LSOs for any purpose that might have privacy implications. Even assuming a pessimistic worst case where all websites with unique identifiers in LSOs are using them to track users, the percentage of such sites studied is low — 9% of the top 100, and only 3.4% of the randomly-selected 500 sites we studied. However, over 40% of the LSOs in each data set used unique identifiers, and especially with the top 100 sites, many people could be affected. Because we found sites using LSOs as unique identifiers, we believe further study is needed to determine if these sites are using LSOs to evade users' choices. However, without visibility into back-end databases, it is difficult to determine how unique identifiers are used. We conclude our paper with policy options and a discussion of implications for industry self-regulation of Internet privacy.

*lorrie@cmu.edu

1 Introduction

Adobe sells several products related to Flash technologies. Some of Adobe's customers are currently being sued for using Flash to store persistent data on Internet users' hard drives, allegedly contrary to users' knowledge after users have deleted HTTP cookies, as a way to bypass users' privacy choices [36]. These lawsuits follow research performed in 2009 by Soltani, et al [31] that found companies using Flash to engage in questionable practices. In this paper we measure the prevalence of "respawning" deleted HTTP cookies, as well as examine the potential for user data to persist beyond deleting HTTP cookies without respawning. We review related work in section 2 and describe our methods in section 3. We present our findings in section 4. We discuss policy implications in section 5, policy options in section 6, and conclude in section 7.

2 Background and Related Work

Flash is used to create multimedia applications, including interactive content and animations embedded into web pages. Flash Player is not natively built into web browsers, but rather is a plugin that works across multiple operating systems and all of the most popular web browsers, allowing developers to easily create cross-platform programs. An estimated 99% of desktop web browsers have the free Flash Player plugin enabled [2].

Early versions of Flash did not allow for direct access to HTTP cookies [32]. Although programs written to run in Flash Player could not read and write HTTP cookies directly, Flash programmers could use an additional programming language, such as JavaScript, to access HTTP cookies [5]. However, using a second language to save and read data was cumbersome and frustrating to Flash developers [8]. As applications written to run in Flash Player evolved beyond playing videos and became more interactive, there were more types of data to save. This is a familiar pattern: web browsers also initially had no way to save state, which was fine when the web was static text and images, but caused limitations as web applications became more complex. Netscape engineers introduced HTTP cookies as a way to support online shopping carts in 1994 [13]. Flash MX was released in 1996, prior to Adobe's purchase of the company that owned Flash. The Flash MX release introduced an analog to HTTP cookies. Adobe refers to this storage as Flash Player Local Shared Objects ("Flash Player LSOs" or just "LSOs"). Flash Player LSOs are commonly referred to as "Flash cookies." Other Internet technologies use local storage for similar purposes (e.g. Silverlight, Java, and HTML5). Although Flash developers could use HTTP cookies to save local data, there are several reasons why Flash developers generally prefer using LSOs, including:

- Flash programmers find LSOs are much easier to work with and write code for than HTTP cookies.
- While JavaScript is built into all major browsers, a small percentage of users choose to disable JavaScript, which would break any applications written to run in Flash Player that relied upon JavaScript to access HTTP cookies.
- LSOs hold more data and support more complex data types than HTTP cookies, giving developers more flexibility and control over what can be stored locally.

See Table 1 for a summary of some of the differences between HTTP cookies and LSOs. Aside from technical differences, HTTP cookies and LSOs are often used to perform the same functions. However, users interact with HTTP cookies and LSOs in different ways. Most users do not fully understand what HTTP cookies are but at least they have heard of them; few users have heard of LSOs [16]. Users have access to HTTP cookie management though browsers' user interfaces, but until recently could not manage LSOs via web browsers' native user interfaces. LSO management required either visiting the Macromedia website to set LSOs to 0 kb of storage, which functionally disables LSO storage, or interacting directly through the Flash Player context menu. Web browsers' "private" browsing modes retained LSOs until early 2010, when Adobe added support for InPrivate browsing [38]. Until recently, most Privacy Enhancing Technologies (PETs) designed to help users manage their HTTP cookies did not address LSO management. So long as

Table 1: Technical differences between HTTP cookies and LSOs

	HTTP Cookies	**LSOs**
Where can the data be read?	Just from the browser that set it	From all browsers on the computer
How long does the data last?	Default: until browser closes, but in practice, commonly set to expire after 18 months or many years	Permanent unless deleted
How much data does it hold?	Maximum: 4 KB	Default: 100 KB, but users can choose higher or lower values
Which data types are supported?	Simple Name/Value pairs	Simple and complex data types

persistent LSOs stored innocuous and anonymous data like game high scores, whether the data were stored in HTTP cookies or LSOs was primarily a technical implementation detail. However, LSO use has evolved into areas with privacy implications.

Advertisers use persistent identifiers in HTTP cookies to help them understand a given customer's browsing history. This data is used to build interest profiles for people in interest groups or demographic categories. Advertisers charge premiums to display ads just to people in specific interest profiles. Advertisers also use HTTP cookies to contribute to analytics data about which customers have viewed ads, clicked on ads, and purchased from ads. Analytics data helps advertisers test different approaches to determine if an ad is effective with a particular audience. More importantly, without at least basic analytics, advertising networks would not know how much to charge for ads. Meanwhile, many users prefer not to be tracked and express that preference by deleting their HTTP cookies [16]. Deleting cookies can cause tremendous problems for analytics data based on HTTP cookies, where even a small error rate can result in incorrectly billing thousands of dollars in a single advertising campaign [33].

Advertisers discovered LSOs addressed their data quality problems [4]. LSOs remained untouched even by users who deleted HTTP cookies because many users did not know about LSOs, they do not expire, and it was often difficult for users to delete them (e.g. under Windows, LSOs write to hidden system folders, away from most users' notice or technical ability to delete.) LSOs are cross-browser, eliminating advertisers' problem with HTTP cookies that a single user using two browsers (for example, Internet Explorer and Firefox) is miscounted as two different users.

Rather than write new code to work with LSOs, in some cases advertisers simply used LSOs to identify a user and then re-create ("respawn") that user's previously deleted HTTP cookie data, enabling advertisers to continue to use their existing code base. For example, starting in 2005 United Virtualities sold a product that used LSOs to "restore" deleted HTTP cookies [9]. United Virtualities explained that this was "to help consumers by preventing them from deleting cookies that help website operators deliver better services" [9]. LSOs used to respawn HTTP cookies sounds like the "best practices" description put forward in a W3C document on mobile web use [34]:

> Cookies may play an essential role in application design. However since they may be lost, applications should be prepared to recover the cookie-based information when necessary. If possible, the recovery should use automated means, so the user does not have to re-enter information.

As a technical response to the technical problem of poor-quality analytics data, using LSOs to respawn HTTP data was a good engineering solution. However, problems collecting analytics data are not just a technical glitch: users *intentionally* delete HTTP cookies as an expression of their desire for privacy. Users

had no visible indication that LSOs existed or that HTTP cookies respawned. Users reacted with surprise when they learned that HTTP cookies they had deleted were not actually gone.

Furthermore, LSOs can be used to track specific computers without respawning HTTP cookies. HTTP cookies can contain a unique identifier so websites can tell when a specific computer has visited the site again. LSOs can be used the same way. Even when users delete their HTTP cookies to protect their privacy, unless they also know to manage LSOs, they may still be identified both to first- and third-party websites via unique identifiers in LSOs. From a user's perspective, this is functionally equivalent to respawning: despite deleting HTTP cookies, they are still being tracked. However, not all unique identifiers are used to track specific computers. For example, each song or video clip on a website could be assigned a unique identifier.

LSOs became a topic of interest in 2009 with the publication of Soltani et. al.'s paper investigating the use of LSOs for respawning deleted HTTP cookies and storing data [31]. They found at least four instances of respawning, and over half of the sites they studied used LSOs to store information about users. Several things changed after the Soltani study:

- Public awareness increased. Media attention popularized the study findings (e.g. [30, 14]) and privacy professionals called attention to LSOs (e.g. [27, 28]).
- Corporate practices changed. Quantcast announced they would no longer respawn HTTP cookies [29]. The Network Advertising Initiative (NAI), an industry group active in self-regulation efforts, published guidelines that their member companies must not respawn HTTP cookies. Further, the NAI bars their members from using local storage[1] for behavioral advertising at all [23].
- Tools improved. Some PETs added LSO management [24, 22]. Adobe added support for "private" web browsing [38] and announced they were working with browser vendors to integrate LSO management into browser user interfaces [11].
- Regulators took an interest. The FTC requested more information from Adobe, and Adobe formally commented to the FTC characterizing respawning as a misuse of LSOs [26].

In 2010, the *Wall Street Journal* ran a new series of articles about Internet privacy. The series included findings from a second Soltani-led study of 50 websites' use of LSOs and tracking technologies, using data collected at the end of 2009 [1]. Subsequent to the new media attention, several class action lawsuits alleging misuse of Flash technologies are currently pending [6].

We collected data from July 12 to 21, 2010, approximately one year after the first Soltani study. This was six months after the data collection for the second Soltani study, but prior to the *Wall Street Journal* coverage, and prior to the lawsuits.

This paper provides another data point in the rapidly changing realm of LSOs. We investigated more sites than both of the Soltani studies with a more reproducible protocol, though we did not investigate sites as deeply. We also extend knowledge about Flash practices by investigating a random sample in addition to popular websites where prior studies focused. We found respawning is currently rare but sites still use LSOs as persistent identifiers (less than what Soltani et. al. found, though again we caution we used different methods), which may or may not have privacy implications, as we discuss further below.

[1] E.g. Flash LSOs, Internet Explorer Browser Helper Objects (BHOs), Microsoft Silverlight objects, etc.

3 Research Methods

We used two identically-configured computers on two different networks to visit 600 websites, and then we analyzed the LSOs and HTTP cookies those sites set. We investigated two different data sets:

- 100 most popular sites as of July 8, 2010
- 500 randomly selected sites

We created these two data sets based on Quantcast's ranked list of the million most popular websites visited by United States Internet users [25]. Both data sets contain international websites although the sites we visited are primarily US-based.

The 100 most popular sites captures data about the sites users are most likely to encounter. This is the same method Soltani et. al. used in their study [31].[2] Because the most popular sites may not follow the same practices as the rest of the web, we also sampled a random population of 500 sites. We list all websites we visited in the Appendix.

We used two identically-configured Windows laptops (XP Pro, version 2002, service pack 3) with Internet Explorer 7 configured to accept all cookies and reject pop ups. We used the most recent version of Flash Player available at that time, 10.1. Our two laptops were on different computer networks so they would not have similar IP addresses, eliminating IP tracking as a potential confound.

LSOs are stored in a binary format. We used custom code from Adobe to save the contents of each LSO in a text file, which allowed us to automate comparisons of log files rather than open each LSO in a SOL editor. This was strictly a convenience and did not alter the data we collected.

At each site we collected all first-party and third-party cookies and LSOs. We used the protocol described below to gain insights into the use of LSOs as identifiers and as mechanisms for respawning HTTP cookies.

We visited each site in three "sweeps" for a total of nine visits:

- Sweep 1, three visits from laptop A
- Sweep 2, three visits from laptop B
- Sweep 3, three visits from laptop A with the LSOs from laptop B

During each sweep, we conducted three back-to-back visits per site. We copied the HTTP cookies and LSOs after each sweep so we could determine when they had been set. We did not clear cookies or LSOs during these three visits, so the final visit had all HTTP cookies and LSOs. After we completed the three visits per site, we deleted all HTTP and LSOs from system directories and moved on to the next site in the dataset. We conducted a total of three sweeps: a sweep on laptop A, a sweep on laptop B on a different network, and then another sweep on laptop A with LSOs copied over from laptop B.

We collected data from the most popular sites, starting on July 14th on laptops A and B. It took five hours to complete a full sweep for the popular sites and 25 hours to complete a full sweep for the randomly selected sites. We then verified our data and re-visited individual sites as needed due to crashes or caching issues, as we describe at the end of this section. Once we confirmed we had data for all sites on both laptops, we began Sweep 3 for the most popular sites on July 15th. We again confirmed data integrity, and completed data collection for two sites that had caching problems on July 21. For the randomly selected sites, we collected data on laptop A starting July 12, laptop B starting July 16, and the third sweep starting July 18. We completed data collection for three sites that had caching problems on July 19th.

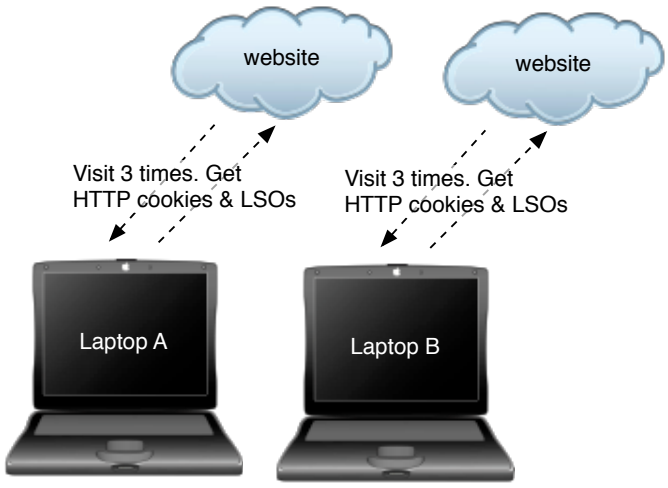
The protocol we followed was designed to contrast content between two different computers, laptops A and B. Any content that is identical on both of the laptops cannot be used for identifying users or computers.

[2] During the course of the year between the first Soltani study and our study, 31 sites that had been in the top 100 in 2009 were displaced with different sites in 2010. In the body of this paper, we present just the top sites from 2010, as there is substantial overlap between the 2010 and 2009 datasets. However, we also studied those 31 sites to be sure they were not substantially different from the 2010 most popular sites. We did not find any additional instances of respawning in the 31 sites that had been in the top 100 sites in 2009 but were no longer in the top 100 in 2010.

For example, one site set the variable testValue to the string test. Every single visitor to that site saves the same string, so there is no way to tell visitors apart because there is nothing unique in the data. On the other hand, a variable holding a unique user id likely identifies a specific computer. For example a site that sets a variable named userID to a unique 32-character string that differs between the two laptops can uniquely identify each of those laptops. In contrast, a site might use a time stamp to note the time the LSO saved to disk. For example a site might set a variable named time to the string 1279042176148 on one laptop, and 1279042395528 on the second laptop. In this case, time stamps are the time elapsed in milliseconds since January 1, 1970. It is not a surprise that the times are slightly different between the two laptops, as we did not start the scripts at exactly the same time. Websites are unlikely to have many visitors at precisely the same millisecond, and can keep the original time stamp indefinitely. While not designed for identification, websites could theoretically use time stamps to distinguish specific computers across multiple visits. However, setting a time stamp is a standard practice. This is one case where variance between laptops does not automatically mean the data is being used to uniquely identify computers. A variable named userID with unique content is more likely to be used to uniquely identify computers than a variable named time. However, we do not have visibility into how variables like userID and time are used, since only data is stored in LSOs. The programs that use the data reside on computers from the company that set the LSO data. We have no ability to inspect how data are used, just to observe the saved data. In summary, we cannot definitely know how data is used in practice, but we can make intelligent suppositions.

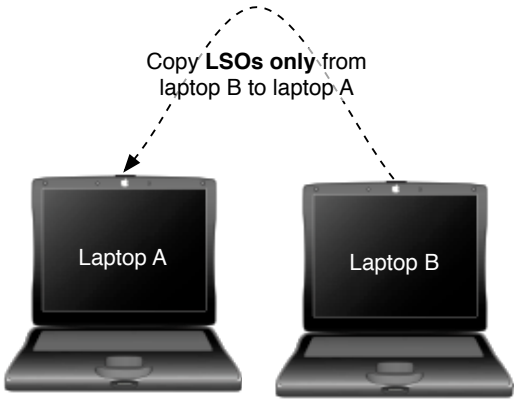
We followed the following automated protocol to collect data for our analysis:

1. Delete all cookies and cached data on both laptops
2. **Sweep 1.** On laptop A, for each site:



   (a) Launch Internet Explorer
   (b) Visit the site
   (c) Wait 60 seconds to allow all cookies to download
   (d) Copy all HTTP cookies, LSOs (*.sol and *.sor) and log files to another directory
   (e) Visit the site two more times to get a rotation of ads and copy all HTTP cookies and LSOs after each visit
   (f) Quit Internet Explorer
   (g) Move all HTTP cookies and LSOs to get any cached files that were saved on exit (deleting all HTTP cookies and LSOs in the process)

3. **Sweep 2.** On laptop B, the exact same procedure as for laptop A in step 2 above.
4. **Sweep 3.** On laptop A, for each site:



   (a) Copy the final set of LSOs only (not HTTP cookies) that had been on laptop B for that site into the ..\Application Data\Macromedia directory on laptop A
   (b) Visit the site just with laptop A

   (c) Wait 60 seconds to allow all cookies to download
   (d) Copy all HTTP cookies, LSOs (*.sol and *.sor) and log files to another directory
   (e) Visit the site two more times to get a rotation of ads and copy all HTTP cookies and LSOs after each visit
   (f) Quit Internet Explorer
   (g) Move of all HTTP cookies and LSOs to get any cached files that were saved on exit (deleting all HTTP cookies and LSOs in the process)

At the end of this procedure we compared HTTP cookies from all three sweeps. To identify respawning, we looked for HTTP cookie strings that were different on laptops A and B in sweeps 1 and 2, but in sweep 3 were identical to sweep 2. This suggests that the information in the HTTP cookie in sweep 3 propagated from the LSOs copied over from sweep 2. In the two cases of respawning that we observed, the text in

HTTP cookies also matched text in LSOs, but not all matches between HTTP and LSOs were indicative of respawning.

See Figure 1 for a graphical depiction of how we classified sites for the popular and randomly selected websites. As shown in Figure 1, first, we looked for sites that saved an LSO in the *#SharedObjects* subdirectory (step 1). We disregarded all of the sites that did not save LSOs. Second, we compared the file structure on laptops A and B to see if we had LSOs from the same sites with the same file names (step 2). If the file names matched on laptops A and B, then we compared the contents of those files (step 4). If the file contents were identical on laptops A and B, there was nothing unique, and the LSOs could not be used to respawn or to identify computers (step 5). If the content in the LSOs differed between laptops A and B, then we classified these as uniquely identifying, though we cannot be certain if computers are being uniquely identified. We further investigated to see if the unique contents within LSOs matched with content in HTTP cookies (step 6). If not, we classified them as having unique content (step 7) but did not have to check for respawning. We performed a final check. We looked at the HTTP cookies from the third sweep, which was performed with LSOs from laptop B, and checked to see if the HTTP cookies on laptop A now matched the LSO data we copied over from laptop B (step 8). If so, we established HTTP cookies were respawned from data stored in LSOs (step 10). If not, we still knew the LSOs had unique content (step 9).

This describes all of the boxes in the classification flow chart except for the case when we did not find the same file name and path for LSOs on laptops A and B (step 3). Despite visiting sites three times in each sweep to catch rotation of content and ads, on some sites we found third-party LSOs from the first sweep on laptop A, but not on laptop B, or vice versa. For example, we might see the file s.ytimg.com/soundData.sol on laptop A but not laptop B. In all but two instances we had already seen third-party LSOs of that type on other sites where the LSO did appear on both laptops. For example, on a different website, we would see s.ytimg.com/soundData.sol on both laptops A and B, allowing us to determine if there was any unique content, and then classify the soundData.sol LSO. After we classified an LSO, we then applied the same classification for sites with that LSO only on one laptop. This method worked well because there are comparatively few third-party companies using LSOs, and we saw the same third party LSOs multiple times across multiple sites. For all first party sites that used LSOs, we found those LSOs saved to both laptops A and B, not just one laptop. We were unable to classify third-party LSOs on only two out of 600 websites.

We did not traverse multiple pages within websites; we only visited the top level of any given domain. As an example of where that would affect results, some sites start with login pages and only have content designed for Flash Player after users login. We did not do any logins or deep links, which means our counts are a lower bound. We also did not interact with any content in Flash Player. This is less of a concern for quantifying Flash respawning, as sites using LSOs for respawning would typically not want to require user interaction before saving LSOs. Similarly, if companies are using LSOs to uniquely identify visitors to their sites, we expect they would do so immediately and not require interaction with content in Flash Player. However, we expect that we undercounted the total number of sites using LSOs. In addition, we only reported persistent LSOs saved, not all LSOs set: we logged several sites that saved LSOs but then deleted them. Transient LSOs cannot be used to uniquely identify computers over time or for respawning, so we do not report those statistics. Finally, we turned on popup blocking in Internet Explorer to reduce caching issues, which could also undercount any LSOs from blocked popups, but popups are not pervasive at this time.

We did observe sporadic issues with cached data. For example, Flash creates a uniquely-named subdirectory under the #SharedObjects directory, something like 8SB5LMVK.[3] When we quit Internet Explorer and removed all #SharedObjects files and subdirectories, the next site to save an LSO would create a new randomly named #SharedObjects subdirectory. However, in approximately 6% of the sites we visited, when we launched a new version of Internet Explorer it would re-create the prior path and save old LSOs from the prior website. To address this issue, we had to re-run data collection for all sites that had a #SharedObjects subdirectory with the same name as the prior site we visited. This appears to be an issue on the web browser side. We were not able to reproduce it reliably, and did not test other web browsers. From a

[3]These unique directory names cannot be used to identify computers because application programmers are unable to access the name of the directory. The directory names are randomly generated for security reasons.

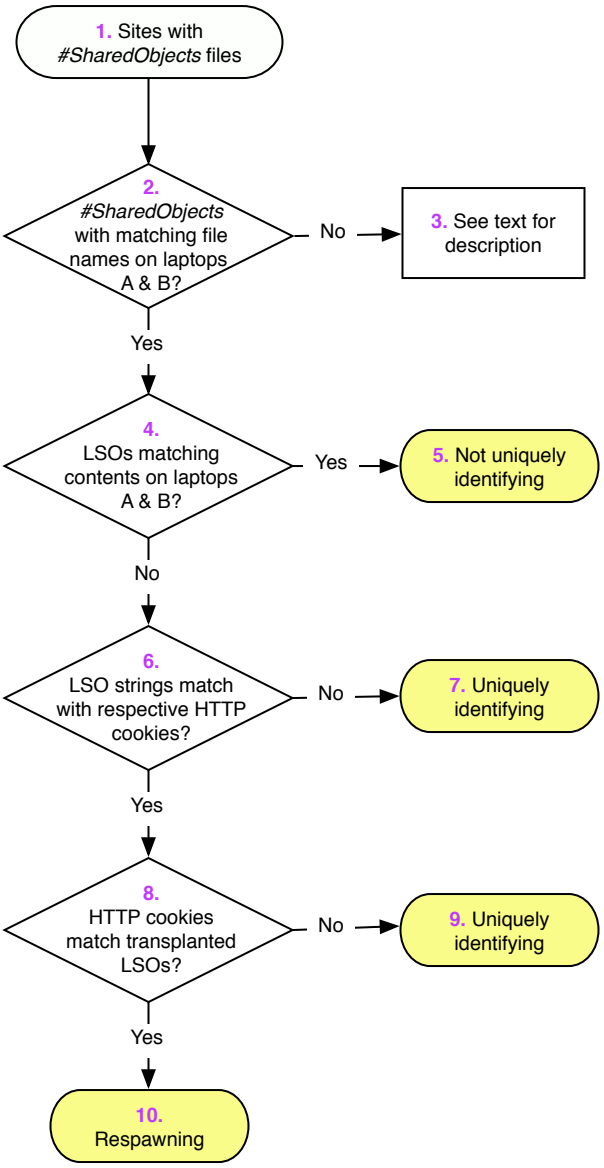


Figure 1: Flow chart of website classification based on SharedObjects. Purple numbers correspond to descriptions in the body of the paper.

user's perspective, cache issues could look like – and function like – respawned LSOs, even though caching issues appear to be completely unintentional.

4 Results

In this section we present our results. First we present our results on the use of HTTP cookies. Then we present our results on the use of LSOs. Overall, we found the most popular sites were more likely to set more HTTP cookies and more LSOs.

4.1 Use of HTTP Cookies

For quantifying HTTP cookie use, there was no advantage to using any particular sweep. We did see a small variation between sweeps, for example the number of sites setting HTTP cookies varied by up to 3% depending on which sweep we used. We used the final sweep for all HTTP cookie counts. In our discussion of the #SharedObjects directory we contrast sweep 1 to sweep 2 to look for unique data. We then check results from sweep 3 to identify HTTP cookie respawning, as described in the prior section.

Cookies are ubiquitous. Only two of the popular sites never used cookies (wikipedia.org and craigslist.org). HTTP cookie use drops to 59% of the random 500 sites. Not only do fewer randomly selected sites use any HTTP cookies, they also set fewer cookies per site than popular sites. We used Internet Explorer, which stores cookies in text files. For example, the list of cookie files from a popular site might look like this:

cupslab@ad.yieldmanager[2].txt
cupslab@www.yahoo[2].txt
cupslab@doubleclick[1].txt
cupslab@yahoo[1].txt
cupslab@voicefive[1].txt

Here we see five different hosts that set cookies (ad.yieldmanager, doubleclick, voicefive, www.yahoo, and yahoo). There is some overlap here — www.yahoo and yahoo are from the same company. But as is the case in this example, in general the number of hosts setting HTTP cookies is roughly equal to the number of different companies setting HTTP cookies on the computer.

The contents of an HTTP cookie file might include something like this:

fpms
u_30345330=%7B%22lv%22%3A1279224566%2C%22uvc%22%3A1%7D
www.yahoo.com/
1024
410443520
30163755
2720209616
30090329
*
fpps
_page=%7B%22wsid%22%3A%2230345330%22%7D
www.yahoo.com/
1024
410443520
30163755
2720209616
30090329
*

In Internet Explorer's implementation each cookie file may contain multiple cookies separated by asterisks. The snippet above shows two different HTTP cookies. The first, fpms, is set to a string that begins u_303... and the second, fpps, is set to a string that begins _page.... Both cookies are served by Yahoo. The remaining data pertains to when the cookies expire and other meta information [19].

As we summarize in Table 2, we found an average of 6.7 HTTP cookie files for the popular sites and 2.5 for the randomly selected sites. We observed a maximum of 34 different cookie files on the popular sites and 30 with the random sites. We found an average of 17 HTTP cookies for the popular sites and 3.3 for the randomly selected sites. We observed a maximum of 92 HTTP cookies set from visiting a single popular site, and a maximum of 73 HTTP cookies from a randomly selected site. Users might be surprised to learn that a visit to their favorite site results in HTTP cookies from dozens of different companies, but this is not a novel finding [31].

Table 2: HTTP Cookies

Data set	% sites with cookies	Avg. # hosts	Max. # hosts	Avg. # cookies	Max. # cookies
Popular	98%	6.7	34	17	92
Random	59%	2.5	30	3.3	73

4.2 Use of LSOs

69% of the popular sites and 33% of the randomly selected sites had some LSO activity, by which we mean they at least created a subdirectory to store LSOs, even if they never actually created any LSOs. 20% of the popular sites stored LSOs in the #SharedObjects directory, as did 8.2% of the randomly selected sites. These are the sites we are interested in as potential sources of either respawning HTTP cookies due to LSOs, or as using LSOs to individually identify computers.[4] We discuss these in more detail below.

We compared the contents of LSOs in #SharedObjects directories on two identically-configured laptops. However, we did not always find identical files on both laptops. For example, one site contained two LSOs on Laptops A and B, but contained an additional two LSOs just on Laptop B.

Six of the 20 popular sites with #SharedObjects did not have matching file names. The random 500 sites include 41 sites with with #SharedObjects, of which nine did did not have matching file names. In both datasets we observed one LSO that we saw only once, so we were unable to classify it.

Why do we see so many mismatches between the two laptops? First party #SharedObjects remained stable. Third party #SharedObjects come from advertisers, and advertising rotates. Even though we collected data on both laptops only a few days apart, advertising — and advertising partners — can change over the course of a few minutes.

4.3 Matched Sites

We found paired LSOs with matching file names on 14 of the 2010 top 100 sites and 32 of the random 500 sites. As mentioned before, any LSO that set identical content on both laptops could not use that content to uniquely identify computers or for respawning. Not all unique identifiers are used for identifying computers, but all identification via LSOs requires a unique identifier. We found matching content on both laptops for six of the 100 popular sites and twenty of the 500 random sites. These sites are neither identifying computers nor respawning. See Figures 2 and 3 for a combined analysis of LSOs with matching file names in all sweeps, as well as LSOs we classified based on seeing them in other contexts.

[4]Programs running in Flash Player also write to the sys directory. While these files are LSOs with the same file format as in the #SharedObjects directory, the sys files are settings that applications programmers cannot edit. There is no API to access the data stored in sys files. Consequently, we have no reason to believe settings files in sys are used for unique identification or respawning.

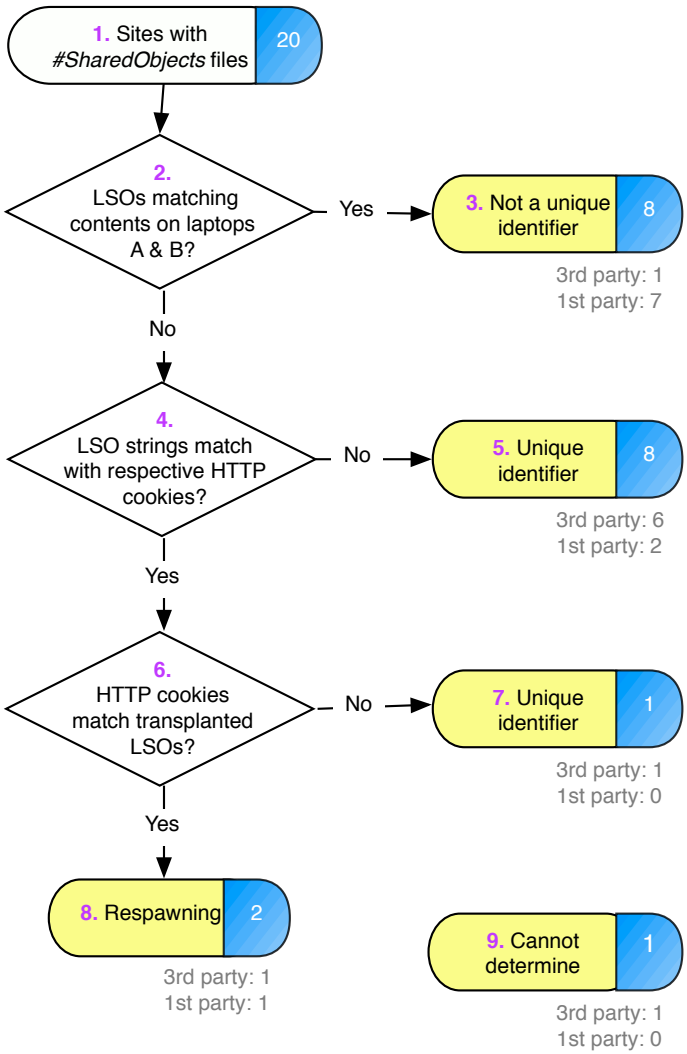


Figure 2: Analysis of the 100 most popular websites in 2010. Blue semi-circles contain the number of sites that fall into a given category. Purple numbers correspond to descriptions in the body of the paper.

4.4 Mismatched Sites

Variable names like userId helped us theorize that many LSOs are used to identify computers, rather than identifying creative content. Without knowledge of back-end practices we cannot determine why LSOs contain unique identifiers, only to quantify how many do. We further investigated to see if content in LSOs matched content in HTTP cookies. If so, we performed analysis to see if respawning occurred, where LSOs are used to reinstate data after a user has deleted an HTTP cookie. For example, we found one LSO that contains a variable named uID set to a unique a 10 digit integer. After we deleted all HTTP cookies and migrated LSOs from one laptop to the other and then revisited the site, the same 10 digit integer now appears in the new HTTP cookies in the final sweep. This is a clear-cut case of respawning.

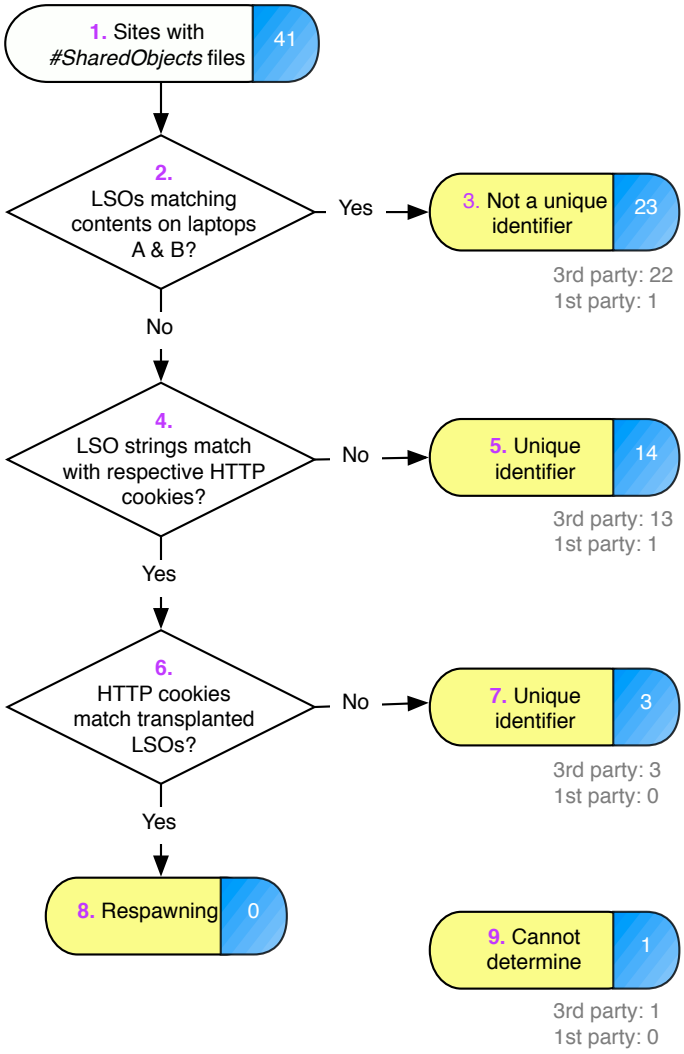


Figure 3: Analysis of the 500 randomly selected websites. Blue semi-circles contain the number of sites that fall into a given category. Purple numbers correspond to descriptions in the body of the paper.

4.5 Prevalence of Unique Identifiers and Respawning in LSOs

As shown in Figure 2, out of 100 popular sites, 20 saved LSOs in the #SharedObjects directory (see oval 1 in Figure 2). Of those 20, eight were not unique content and could not be used for identifying computers or respawning LSOs, and seven of those eight were first-party LSOs (3 in Figure 2). Another nine had unique content and may (or may not) be used to identify computers. Seven of those nine were third-party LSOs (5 & 7 in Figure 2). Two LSOs respawned deleted HTTP cookie content, with one set by a first-party and one from a third-party (8 in Figure 2). We were unable to classify one third-party LSO (9 in Figure 2).

As shown in Figure 3, out of 500 randomly selected sites, 41 saved LSOs in the #SharedObjects directory (see oval 1 in Figure 3). Of those 41, 23 were not unique content and could not be used for identifying computers or respawning LSOs, and 22 of those 23 were third-party LSOs (3 in Figure 3). Another 17 had unique content and may (or may not) be used to identify computers. Sixteen of those 17 were third-party LSOs (5 & 7 in Figure 3). We observed no respawning in the random 500 dataset (8). We were unable to classify one third-party LSO (9 in Figure 3).

4.6 Response to Respawning

In October, 2010, the Center for Democracy and Technology (CDT) attempted to contact the two sites we found were respawning HTTP cookie content from LSOs. CDT successfully contacted one site, where site operators expressed surprise to learn they were respawning LSOs. The site voluntarily stopped using LSOs while they conducted an internal review. In subsequent discussions with CDT, they stated they were not using LSOs for respawning. They were counting unique visitors to their site. At this time, they no longer use unique identifiers in LSOs for analytics. We have visited the site multiple times, and confirmed the site no longer sets LSOs.

CDT was unable to reach the third-party company that respawned HTTP cookies at the second site. CDT left messages by voice mail and email describing concerns with respawning in mid-October. However, even before CDT's messages, this company stopped respawning cookies by August 30th on the first-party site we studied. We did still see HTTP cookies from the third-party on September 14th, which establishes they still had a relationship with the first-party site and it was not simply a case that they stopped doing business together. Furthermore, CDT created a list of companies that had a relationship with this third-party company based on the contents of their website, blog posts, and news articles. CDT visited all of those sites and found no LSOs from the third-party company that had been respawning.

CDT left messages for companies that use LSOs to set unique identifiers. We hoped to understand to what extent unique identifiers were used to uniquely identify computers, rather than for a non-tracking purpose. None of the companies CDT attempted to contact were willing to speak with CDT.

We subsequently analyzed the privacy policies for the companies setting unique identifiers to see if we could determine their practices based on their privacy policies. For the eight popular sites with unique identifiers, their policies were unclear and we were not able to determine if they use LSOs to uniquely identify specific computers.

For the random sites, we looked at both the first-party website and any third-parties setting an LSO, for a total of 32 unique sites. Once again, we were unable to determine if any of the sites use LSOs to uniquely identify specific computers.[5]

Finally, we reviewed the privacy policies for the two first-party websites where we found respawning, plus the third-party website engaged in respawning. The first-party websites' privacy policies were unclear. The third-party did not have a privacy policy.

[5]Of those 32 sites, 14 sites (44%) did not have privacy policies, including one site that was taken offline by law enforcement agents. None of the sites made promises that would be violated if they use LSOs to uniquely identify computers. None of the sites stated that they use LSOs to uniquely identify specific computers. Four of the sites (13%) gave hints that they might be using LSOs to uniquely identify specific computers, for example discussing "cookies and other means," to re-identify visitors to the sites, or disclosing LSO use to combat fraud and for "other purposes." The remaining 18 sites (44%) had policies that were completely unclear or did not mention LSOs at all. In all, we were able to neither definitively classify any of the sites as using LSOs to identify individual computers, nor able to definitively rule it out.

5 Policy Implications

While our results suggest that use of LSOs to respawn HTTP cookies or track users may be declining, the frequent presence of unique identifiers in LSOs combined with a lack of transparency about the use of these LSOs continues to raise concern. But using LSOs to track users is just the tip of the iceberg: new mechanisms continue to emerge that are designed to track users in ways that circumvent privacy controls [35].

HTTP cookie respawning generated media attention and regulatory interest. In part, this may be because respawning implies such a blatant disregard for user choice. More subtle practices with similar functionality are just as dangerous to privacy, but may not be as clear-cut topics for regulatory authority. In this section we briefly address a few points that pertain not just to LSOs and respawning, but to the larger topic of Internet privacy.

First, regulators are likely to reject industry self-regulation if even the most prominent companies will not respect user choice. It is difficult to find calls for a purely industry self-regulation approach to Internet privacy credible when industry demonstrates willingness to violate user intent and privacy as demonstrated by using LSOs to respawn HTTP cookies or individually identify computers. No malice is required: it is easy to imagine software engineers using a clever tactic to avoid expensive data loss without considering privacy implications. But the effects on user privacy are the same regardless of how decisions are made.

Second, when the Center for Democracy and Technology cannot get companies to answer questions about their privacy practices, and privacy researchers cannot determine privacy practices by reading privacy policies, it seems unreasonable to expect end users to be able to understand when LSOs are being used and in what capacity. One of the appealing features of an industry self-regulation approach is that because privacy preferences vary greatly between individuals, self-regulation allows users to choose what is appropriate for them personally. However, what we see in this case is that users lack the information to make choices. Absent better communication, privacy policies cannot form the basis of informed consent.

Third, one of the arguments against legislative or regulatory action with regard to the Internet is that companies can innovate faster than government can respond. That is likely true in some contexts. However, because companies can move quickly does not mean they will move quickly, particularly when action is against their economic interests. To draw on an example specifically from this context, a representative from Macromedia (developers of Flash technologies, acquired by Adobe) responded to privacy concerns saying that they did not think Flash Player was a privacy threat, but they were speaking with browser makers to improve LSO management — in 2005 [7]. That LSO management was not addressed until it became a crisis five years later does not seem unusual. Any software team prioritizing what to work on for the next release will have a hard time arguing for a theoretical threat to privacy as something to address before adding new features that could sell more of their product or fixing bugs that annoy their current user base. When multiple companies work together (i.e. Adobe and browser companies) delays are even more likely than when companies are able to act independently. In the context of Internet privacy, government moving slowly may still bring more progress than companies will make on their own.

Fourth, a common mental model of user choice for privacy is that users can decide which HTTP cookies to accept, or decide to delete specific HTTP cookies. With a single site setting over 90 cookies this concept is outdated. No one can practically choose yes or no for each HTTP cookie when there are so many of them in use. As LSOs and other technologies are being used for tracking, user control becomes even more difficult. In order to manage HTTP cookies users must rely on some type of privacy enhancing technology even if it as simple as settings in their web browser. Other options for HTTP cookie management exist, including stand-alone packages like CCleaner, opt-out cookies, and browser plugins. We have crossed the threshold where users require PETs if they are to protect their online privacy.

Finally, the proposed Best Practices Act would create a safe harbor for companies working with the FTC, while other companies would still be subject to lawsuit. Opponents are concerned that privacy lawsuits would only enrich trial lawyers while proponents argue the threat of lawsuit would improve practices [10]. While lawsuits are a cumbersome and inherently reactive approach to privacy, we did see possible support

for the view that the threat of lawsuit can improve practices. In particular, we note the third-party company that we observed respawning. They stopped respawning after media coverage of lawsuits, but before we contacted them. That they would not answer voice mail or email also suggests they may have been wary of legal action. Furthermore, the sites identified as respawning in both of the Soltani studies appear to have stopped respawning. Our experience is not conclusive, but may be worth considering.

6 Policy Options

In this section we examine which stakeholders can take steps to reduce privacy-sensitive LSO practices. It is an open question how many resources should be expended. Our results suggest that problems with LSOs are reducing over time, but are still present. However as noted in the previous section, LSO abuse is only one element of a larger problem. Ideally, policy solutions do not address technologies one-by-one, but rather address the entire class of technologies used to track users without informed consent. That said, here are some steps that stakeholders could take to address LSOs.

6.1 Companies Using Flash Technologies

The ultimate responsibility for using LSOs to respawn HTTP cookies rests with the companies that engage in such practices. Unfortunately, even prominent companies have engaged in respawning. We believe, but cannot definitively prove, that additional prominent companies are using LSOs to identify users without respawning.

While these stakeholders are in the best position to take direct action, they benefit from improved analytics and other user data. They are unlikely to change their practices without external motivation. We also note that companies are not always aware when they are using LSOs to respawn HTTP cookies. Chief Privacy Officers (CPOs) or other appropriate staff might visit their own websites to understand if and how they use LSOs. By doing so, CPOs can help their companies avoid potential litigation, regulatory interest, and negative press.

6.2 Adobe

While Adobe did not create privacy problems with LSOs, they inherited the potential for issues when they acquired Flash technologies. Adobe is in a pivotal position to affect Flash developers. Adobe has already taken some actions, including their statement that respawning is abuse of LSOs. However, they have not published a position on using LSOs to uniquely identify computers without respawning HTTP cookies. Adobe could take a stance similar to the IAB position that LSOs must not be used for behavioral advertising at this time, or go beyond that to also include analytics. More generally, Adobe could adopt the policy that LSOs should only be used to support Flash content and nothing else. We do not offer opinions on where Adobe should set their policy, but these seem like some obvious additions to consider and discuss.

Adobe's statement that respawning constitutes abuse of LSOs may not be widely understood by Flash developers, and currently lacks any threat of enforcement. Adobe could communicate their policies clearly in all developer documentation, terms of service, and in popular developer fora. Adobe could also choose to follow Facebook's example and rescind licenses for companies that do not delete inappropriately collected data and do not comply with Adobe's license terms [37]. This is, by nature, an after-the-fact remedy that would only affect companies that have been shown to engage in unacceptable practices, and is not a panacea.

Adobe is currently working to improve users' ability to manage LSOs. They are taking two approaches: working with web browser companies, and redesigning the user interface for controls currently built into Flash. In working with web browsers, Adobe published an API for use with Netscape Plugin Application Programming Interface (NPAPI) [21]. Most web browsers use NPAPI with the notable exception of Internet Explorer [18], necessitating another approach. Adobe touts benefits for security and sandboxing, but their preliminary announcement did not mention privacy [3]. By focusing just on security, Adobe may not have

clearly communicated to the Flash developer community that privacy issues are a priority. In January, 2011, Adobe announced details of interim user interface controls and discussed them in the context of privacy [11].

Flash developers may not think about privacy concerns while in the midst of trying to get code to work. Adobe could add text about privacy to the ActionScript API documentation. Specifically, it might help to add information about acceptable practices to the SharedObject API, which documents how to set and use LSOs. Adobe could also help the Flash developer community by adding a chapter specifically about privacy, to mirror the security chapter in the ActionScript Developer's Guide.

Adobe could modify the functionality of LSOs, but that may risk breaking existing content designed for Flash Player for the majority of developers who have done nothing untoward. This is a difficult issue. To minimize compatibility issues, it is often easier to add new fields than to delete or modify existing fields. For example, in future versions of Flash Player, all LSOs could have an expiration date. This would not prevent LSO abuse, but could limit the scope of privacy issues, and is in keeping with HTTP cookies.

6.3 Browser Companies

Asking browser makers to expend engineering resources for problems they did not create seems unsatisfying, but they do have the ability to improve user experience. LSOs are only one of many types of tracking technologies and browser vendors may need to keep adjusting to prevent new approaches from being used to track users without users' knowledge.

One challenge browser companies face is creating usable interfaces. Users currently struggle to understand how to manage their HTTP cookie preferences [15, 17]. As browser interfaces expand to include managing other types of persistent storage, including LSOs, browser companies have the opportunity to improve the usability of their privacy settings. If browser companies simply tack on other types of storage to their sometimes obscure HTTP cookie management settings, they are likely to increase users' confusion.

6.4 Policy Makers

Focusing specifically on the technology of respawning just creates incentives for developers to move to other types of tracking. As we have mentioned LSOs can store unique identifiers that are functionally equivalent to respawning. The company Mochi Media offers tracking via ActionScript code embedded into content running in Flash Player, with no need to respawn HTTP cookies [20]. A popular book on analytics includes directions on how to use Flash technologies to track what users read in the New York Times, even from mobile devices that are disconnected from the web at the time [12]. These examples happen to be about Flash technologies, but could just as easily be about JavaScript, super cookies, browser fingerprints, or iPhone and iPad unique identifiers. Rather than a narrow focus on specific technologies, policy makers would be well advised to look at functionality.

For enforcement, it seems sensible to focus on the most popular websites. Not only do they reach millions of people, we found they are more likely to have questionable privacy practices. If enforcement actions become public, large companies are more likely to draw press attention than small companies, which will help educate website developers that there are privacy issues they need to consider.

7 Conclusions

We found that while companies were still respawning HTTP cookies via LSOs as late as July, 2010, the number of companies involved was low. We observed HTTP cookie respawning on the front page of only two of the top 100 websites and none of the randomly selected 500 websites we checked. Further, both companies that were respawning have stopped this practice, one on their own, and one as a result of this study. However, because the sites that had been respawning are very popular, many users may have been affected by even just two companies respawning, though respawning is by no means endemic at this time.

Further, we found sites using LSOs to set unique identifiers. While we cannot know definitively how these identifiers are used in practice, we believe some of them identify individual computers. If so, this is functionally equivalent to respawning HTTP cookies. Companies may use LSOs to track users who decline or delete HTTP cookies, but do not realize they also need to manage LSOs. We observed fairly low rates of LSOs that may be identifying computers, 9% for the most popular 100 websites, and 3.4% of a random selection of 500 websites. However, again, the most popular sites reach a very large number of users so many people may be affected by these practices. Furthermore, a little over 40% of sites that save LSO data store unique identifiers, suggesting that Flash developers may not understand LSOs as a privacy concern.

Finally, we note that the most popular sites are more likely to engage in practices with potential privacy implications. We observed primarily third-party LSOs in the randomly selected 500 websites, which again suggests it is possible to work with a small number of prominent companies to dramatically affect practices, rather than needing to contact a large number of small companies. We have hope that LSO use to circumvent users' privacy preferences can be reduced, but note that many other technologies exist that will fill the same function. So long as we focus on individual technologies, rather than a larger picture of user privacy and control, we risk an arms race with advertisers changing the technologies they use to identify users, regardless of users' privacy preferences.

8 Acknowledgments

Support for this project was provided in part by Adobe Systems, Inc. Thanks to Adobe and the Center for Democracy & Technology for their assistance in developing the experimental protocol. Thanks to Justin Brookman, D. Reed Freeman, Kris Larsen, Deneb Meketa, Erica Newland, Gregory Norcie, MeMe Rasmussen, Ari Schwartz, and Peleus Uhley for providing assistance and feedback.

A Appendix

We analyzed two data sets based on Quantcast's list of the million most visited websites: the 100 most visited sites in the United States as of July 2010 and 500 sites we randomly selected from the Quantcast list of one million. We list those sites here.

Table 3: Quantcast's top 100 most visited websites as of July 8, 2010

about.com	adobe.com	amazon.com
americangreetings.com	answers.com	aol.com
ap.org	apple.com	ask.com
associatedcontent.com	att.com	bankofamerica.com
bbc.co.uk	bestbuy.com	bing.com
bizrate.com	blinkx.com	blogger.com
blogspot.com	bluemountain.com	break.com
careerbuilder.com	causes.com	chase.com
chinaontv.com	city-data.com	cnet.com
cnn.com	comcast.com	comcast.net
craigslist.org	dailymotion.com	digg.com
drudgereport.com	ebay.com	ehow.com
evite.com	examiner.com	facebook.com
flickr.com	formspring.me	go.com
godaddy.com	google.com	hp.com
hubpages.com	huffingtonpost.com	hulu.com
ign.com	imdb.com	latimes.com
legacy.com	linkedin.com	live.com
mapquest.com	match.com	merriam-webster.com
metacafe.com	microsoft.com	monster.com
msn.com	mtv.com	mybloglog.com
myspace.com	netflix.com	nytimes.com
optiar.com	pandora.com	paypal.com
people.com	photobucket.com	reference.com
reuters.com	simplyhired.com	suite101.com
target.com	thefind.com	tmz.com
tumblr.com	twitpic.com	twitter.com
typepad.com	usps.com	walmart.com
washingtonpost.com	weather.com	weatherbug.com
webmd.com	wellsfargo.com	whitepages.com
wikia.com	wikipedia.org	windows.com
wordpress.com	wunderground.com	yahoo.com
yellowpages.com	yelp.com	youtube.com
zynga.com		

Table 4: Random selection of 500 sites

24hourpet.com	350smallblocks.com	411webdirectory.com
72712.com	787787.com	aalas.org
aartkorstjens.nl	abbottbus.com	accutronix.com
ad-mins.com	adaholicsanonymous.net	adamscountyhousing.com
adorabubbleknits.com	advanceexpert.net	agnesfabricshop.com
air-land.com	alignmed.com	allstarsportspicks.com
almostfrugal.com	amandabeard.net	amazingamberuncovered.com
amigofoods.com	ancestryhost.org	appcelerator.com
ar-10-rifles.com	arcadianhp.com	archerairguns.com
ariionkathleenbrindley.com	arizonabattery.com	arizonahealingtours.com
asbj.com	asiainc-ohio.org	askittoday.com
askmd.org	asla.org	astonhotels.com
atbfinancialonline.com	athenscountyauditor.org	auburncountryclub.com
auctioneeraddon.com	autorepairs-guide.info	avistarentals.com
awildernessvoice.com	azbiz.com	babygotfat.com
backwoodssurvivalblog.com	badvoter.com	bargainmartclassifieds.com
battlestargalactica.com	beaconschool.org	beatport.com
beechwoodcheese.com	benedictinesisters.org	best-hairy.com
bestshareware.net	bethpage.coop	bf1systems.com
bibleclassbooks.com	bibleverseposters.com	bird-supplies.net
blackopalmine.com	bladesllc.com	blogmastermind.com
bluetoothringtones.net	body-piercing-jewellery.com	bookjobs.com
boulevardsentinel.com	boyntonbeach.com	bradcallen.com
brealynn.info	brill.nl	broncofix.com
buckstradingpost.com	bucky.com	buyhorseproperties.com
bwcnfarms.com	cabands.com	cabins.ca
cafemomstatic.com	capitalgainsmedia.com	cardiomyopathy.org
careerstaffingnow.com	carrollshelbymerchandise.com	cashloanbonanza.com
cateringatblackswan.com	cdcoupons.com	charterbank.com
charterco.com	chashow.org	cheapusedcars.com
childrensheartinstitute.org	christmas-trees-wreaths-decorations.com	clarislifesciences.com
claytonihouse.com	clcofwaco.org	clean-your-pcc1.com
cloningmagazine.com	clubdvsx.com	codeproject.com
coltbus.org	coltranet.com	columbusparent.com
complxregionalpainsyndrome.net	computervideogear.com	conservativedvds.com
cookbooksforsale.com	coolatta.org	corvettepartsforsale.com
countrymanufacturing.com	cpainquiry.com	crazyawesomeyeah.com
crbna.com	creatupropiaweb.com	credit-improvers.net
creditcaredirect.com	crowderhitecrews.com	culttvman2.com
curepeyronies.net	curiousinventor.com	dansdidnts.com
dardenrestaurants.com	datingthoughts.com	dcso.com
de.ms	dealante.com	dealsoutlet.net
delti.com	desktops.net	detroitmasonic.com
digitalmania-online.com	disasterreliefeffort.org	dividend.com
dmvedu.org	dobbstireandauto.com	dodgeblockbreaker.com
donlen.com	donnareed.org	dorpexpress.com
dukeandthedoctor.com	dvdsetcollection.com	easypotatosalad.com
educationalrap.com	elmersgluecrew.com	emailfwds.com
emailsparkle.com	empty.de	ereleases.com
escapethefate.net	eurekasprings.org	evanity.com
expowest.com	eyesite.org	fashionreplicabags.com
fast-guardcleaneronpc.net	fatlove.net	fearrington.com
fitnesshigh.com	flatpickdigital.com	fleetairarmarchive.net
florahydroponics.com	floridafishinglakes.net	flyingbarrel.com
foodtimeline.org	foreclosuredlist.com	foreclosurepulse.com
forzion.com	fourreals.com	free-party-games.com
freepetclinics.com	freshrewardscore.com	fretwellbass.com
fukushima.jp	fullertontitans.com	fundmojo.com
fusioncrosstraining.com	ga0.org	gaara.ws
ganstamovies.com	gemission.org	genesearch.com
gerdab.ir	getanagentnow.com	girlfights.com
globalfire.tv	gmeil.com	gogivetraining.com
gold-speculator.com	goldenstaterails.com	gomotobike.com
goodseed.com	googgpillz.com	gordonbierschgroup.com
gotostedwards.com	goutresource.com	graceandtruthbooks.com
grooveeffect.com	hairybulletgames.com	hallfuneralchapel.com
hallmarkchannel.tv	hammondstar.com	happyshoemedia.com
healthcaresalaryonline.com	hills.net	historyofnations.net
hoover-realestate.com	horseshoes.com	hostpapa.com
hoveringads.com	howyouspinit.com	hp-lexicon.com
hsbc.com.mx	hvk.org	icdri.org
idxcentral.com	ieer.org	iflextoday.com
indianapolis.com	infinitiofdenver.com	inhumanity.com

Continued on Next Page...

Table 4 – Continued

inria.fr	intelos.com	iphonealley.com
iris-photo.com	itmweb.com	itvs.com
itw.com	ivanview.com	jacksoncountygov.com
japanautopages.com	jesus-passion.com	jetbroadband.com
jimmycanon.com	josejuandiaz.com	joybauernutrition.com
junohomepage.com	jwsuretybonds.com	kbduct.com
kimballarea.com	kitten-stork.com	knittingpureandsimple.com
kpcstore.com	lacosteshoes.us	lafarge-na.com
lakeareavirtualtours.com	latinrank.com	layover.com
life-insurance-quotes-now.com	lifepositive.com	liftopia.com
like.to	lintvnews.com	logodogzprintz.com
lstractorusa.com	ltwell.com	lydiasitaly.com
madisonindiana.org	magnetworks.com	marketminute.com
mastiffrescue.org	maurywebpages.com	mayoarts.org
mcpherson.edu	mcswain-evans.com	measurebuilt.com
meiselwoodhobby.com	menalive.com	merbridal.com
michiganford.com	microcenter.com	miltonmartintoyota.com
minki.net	mirdrag.com	missourimalls.net
mistercater.com	mitutoyo.com	mmodels.com
modbee.com	moforaja.com	moldingjobs.com
moneytip.com	moselhit.de	motomatters.com
motosolvang.com	movefrontlistencom.com	mule.net
mundofree.com	my-older-teacher.net	mycomputerclub.com
mylexia.com	mypickapart.com	mystic-nights.com
mysticalgateway.com	mysticlake.com	mytableware.com
nationalcoalition.org	naturalmedicine.com	ncbeachbargains.com
ncgold.com	nec.jp	nekoarcnetwork.com
newcracks.net	newlawyer.com	newmacfurnaces.com
newscoma.com	nexstitch.com	nhlottery.com
nittygrittyinc.com	nobledesktop.com	nottslad.com
npg.org.uk	nscale.org.au	nwlanews.com
ocharleydavidson.com	offscreen.com	oixi.jp
olympus-imaging.com	omahaimpound.org	onelasvegas.com
onepaycheckatatime.com	optimost.com	orchidphotos.org
outbackphoto.com	ownacar.net	ownthenight.com
p2pchan.info	parkcityinfo.com	parksandcampgrounds.com
paulrevereraiders.com	pedalmag.com	pennhealth.com
performancehobbies.com	perthmilitarymodelling.com	pet-loss.net
petworld.com	pgamerchandiseshow.com	planfor.fr
plantronics.com	pngdealers.com	polapremium.com
policespecial.com	pphinfo.com	promotersloop.com
promusicaustralia.com	prophecykeepers.com	prostockcars.com
psychprog.com	puppyluv.com	puppystairs.com
q102philly.com	qdobamail.com	quickappointments.com
quickertek.com	quickfinder.com	raleyfield.com
raphaelsbeautyschool.edu	rareplants.de	rax.ru
readingequipment.com	realtracker.com	rentonmclendonhardware.com
restaurantsonlinenow.com	resveratrol20.com	reu.org
revengeismydestiny.com	ripcordarrowrest.com	rpmrealty.com
rrrmusic.com	rumc.com	russellrowe.com
russianbooks.com	sacramentoconventioncenter.com	salonhogar.net
santaslodge.com	scalemodeltoys.com	scanner-antispyh4.com
sccmo.org	scgsgenealogy.com	scottpublications.com
sdchina.com	search4i.com	searchgenealogy.net
section4wrestling.com	seelyewrightofpawpaw.net	seewee.net
sheisladyboy.com	shipleydonuts.com	shootangle.com
shouldersurgery.org	simcomcity.com	simplesignshop.com
socalmls.com	sohojobs.org	southwestblend.com
spanderfiles.com	spatechla.com	squireparsons.com
srtk.net	standup2cancer.org	start-cleaning-business.com
statenotary.info	stimuluscheck.com	stjosephccschool.net
stmaryland.com	storagedeluxe.com	stranges.com
sud.org.mx	sudzfactory.com	summer-glau.net
sungardpsasp.com	sureneeds.com	sweetdealsandsteals.com
sweettattianna.com	swingstateproject.com	syque.com
tackletog.com	tamusahr.com	tasteequip.com
tecnocino.it	tempgun.com	texasthunder.com
the-working-man.com	theacademic.org	theacorn.com
theauctionblock.org	thedailymaverick.co.za	thedigitalstory.com
theelator.com	thegardenhelper.com	thegriddle.net
thegunninghawk.com	theinductor.com	theliterarylink.com
themainemarketplace.com	themodelbook.com	thenextgreatgeneration.com
thepromenadebolingbrook.com	therichkids.com	threebarsranch.com
thunderracing.com	tickledpinkdesign.net	tj9991.com

Continued on Next Page...

Table 4 – Continued

todayswebspecial.com	top-forum.net	toponlinedegreechoices.com
tracksideproductions.com	trafficinteractive.com	transfermarkt.de
treadmillstore.com	tri-une.com	tropicalfishfind.com
trycovermate.com	ttsky.com	twaa.com
twtastebuds.com	ualpaging.com	uniquetruckaccessories.com
univega.com	unon.org	uprius.com
usaplforum.com	uscoot.com	v-picks.com
vacuumtubeonline.com	valueoasis.com	vandykerifles.com
vcbank.net	vet4petz.com	vidaadois.net
videocelebs.org	visitshenandoah.com	vitamin-supplement-reference.com
vitruvius.be	walmartdrugs.net	wcha.org
weddingnet.org	wefong.com	wegotrecords.com
weplay.com	wetzelcars.com	wi-fihotspotlist.com
wiara.pl	wildfoodadventures.com	willyfogg.com
windsorhs.com	wippit.com	womantotal.com
woodauto.com	woodenskis.com	woollydesigns.com
woolrichhome.com	worldcrops.org	worldmapfinder.com
worlds.ru	wwwcoder.com	wxc.com
ymcatriangle.org	youthoutlook.org	ywcahotel.com
zabaware.com	ziua.ro	

References

[1] Tracking the trackers: Our method. *Wall Street Journal* (July 31 2010).

[2] ADOBE SYSTEMS. Flash player penetration, 2010. http://www.adobe.com/products/player_census/flashplayer/ Accessed 21 August 2010.

[3] BETLEM, P. Improved Flash Player support in Chrome, March 2010. http://blogs.adobe.com/flashplayer/2010/03/improved_flash_player_support.html.

[4] BOUTIN, P. Flash cookies get deleted, skew audience stats as much as 25 percent. *VentureBeat* (April 2010). http://venturebeat.com/2010/04/14/flash-cookies-get-deleted-skew-audience-stats-as-much-as-25- percent/ Accessed 13 June 2010.

[5] CARR, D. Integrating Flash content with the HTML environment, April 2008. https://www.adobe.com/devnet/dreamweaver/articles/integrating_flash_html.html Accessed 21 August 2010.

[6] CHENG, J. Lawsuit: Disney, others spy on kids with zombie cookies. *Ars Technica* (August 16 2010). http://arstechnica.com/tech-policy/news/2010/08/lawsuit-disney-others-spy-on-kids-with-zombie-cookies.ars Accessed 21 August 2010.

[7] COHN, M. Flash player worries privacy advocates. *Information Week* (2005). http://www.informationweek.com/news/showArticle.jhtml?articleID=160901743.

[8] ELECTRONIC PRIVACY INFORMATION CENTER. Local shared objects – "Flash cookies", July 2005. http://epic.org/privacy/cookies/flash.html.

[9] GONSALVES, A. Company bypasses cookie-deleting consumers. *Information Week* (March 2005).

[10] GROSS, G. Lawmakers hear mixed reviews of web privacy bill. *PCWorld* (2010). http://www.pcworld.com/businesscenter/article/201712/lawmakers_hear_mixed_reviews_of_web_privacy_bill.html.

[11] HUANG, E. On improving privacy: Managing local storage in Flash Player, January 2011. http://blogs.adobe.com/flashplatform/2011/01/on-improving-pr\ivacy-managing-local-storage-in-flash-player.html.

[12] KAUSHIK, A. *Web Analytics 2.0*. Sybex, 2010.

[13] KRISTOL, D. M. HTTP cookies: Standards, privacy, and politics. *ACM Transactions on Internet Technology (TOIT) 1*, 2 (November 2001).

[14] LEYDEN, J. Sites pulling sneaky Flash cookie-snoop. *The Register* (August 2009).

[15] MCDONALD, A., AND CRANOR, L. F. An empirical study of how people perceive online behavioral advertising. Tech. Rep. CyLab Technical Report 09-015, Carnegie Mellon, November 2009. http://www.cylab.cmu.edu/research/techreports/tr_cylab09015.html.

[16] MCDONALD, A. M., AND CRANOR, L. F. Americans' attitudes about internet behavioral advertising practices. In *Proceedings of the 9th Workshop on Privacy in the Electronic Society (WPES)* (October 4 2010).

[17] MCDONALD, A. M., AND CRANOR, L. F. Beliefs and behaviors: Internet users' understanding of behavioral advertising. In *38th Research Conference on Communication, Information and Internet Policy (Telecommunications Policy Research Conference)* (October 2 2010).

[18] MICROSOFT. Netscape-style plug-ins do not work after upgrading Internet Explorer, July 2007. http://support.microsoft.com/kb/303401.

[19] MICROSOFT. HTTP cookies (windows), December 2010. http://msdn.microsoft.com/en-us/library/aa384321%28v=vs.85%2\9.aspx.

[20] MOCHIMEDIA. Flash tracking, traffic monitoring, and analytics service. http://www.mochibot.com/.

[21] MOZILLA. Npapi:clearprivacydata, December 2010. https://wiki.mozilla.org/NPAPI:ClearPrivacyData.

[22] NETTICAT. Add-ons for Firefox: BetterPrivacy, July 2010. https://addons.mozilla.org/en-US/firefox/addon/6623.

[23] NETWORK ADVERTISING INITIATIVE. FAQ, 2010. http://www.networkadvertising.org/managing/faqs.asp#question_19 Accessed 5 March 2010.

[24] PIRIFORM. CCleaner features. http://www.piriform.com/ccleaner/features.

[25] QUANTCAST. Audience measurement, lookalike modeling, audience buying, July 2010. http://www.quantcast.com/top-sites-1.

[26] RASMUSSEN, M. J. Re: Comments from Adobe Systems Incorporated — privacy roundtables project no. p095416, January 2010. http://www.ftc.gov/os/comments/privacyroundtable/544506-00085.pdf.

[27] SCHNEIER, B. Flash cookies, August 2009.

[28] SCHOEN, S. New cookie technologies: Harder to see and remove, widely used totrack you. Electronic Frontier Foundation, September 2009.

[29] SINGEL, R. Flash cookie researchers spark Quantcast change. *Wired August* (2009). http://www.wired.com/epicenter/2009/08/flash-cookie-researchers-spark-quantcast-change/.

[30] SINGEL, R. You deleted your cookies? Think again. *Wired* (August 2009).

[31] SOLTANI, A., CANTY, S., MAYO, Q., THOMAS, L., AND HOOFNAGLE, C. J. Flash cookies and privacy, August 11 2009. http://ssrn.com/abstract=1446862 Accessed 15 Apr 2010.

[32] STACKOVERFLOW. How do I access cookies within Flash?, 2008. http://stackoverflow.com/questions/109580/how-do-i-access-cookies-within-flash.

[33] STORY, L. How many site hits? Depends who's counting. *New York Times* (October 2007).

[34] SULLIVAN, B. Mobile web best practices 2.0: Basic guidelines, W3C editor's draft, March 2008. http://www.w3.org/2005/MWI/BPWG/Group/Drafts/BestPractices-2.0/ED-mobile-bp2-20080327#bp-cookies-recover Accessed 21 September 2010.

[35] TIMMER, J. It is possible to kill the evercookie. *Ars Technica* (October 2010). http://arstechnica.com/security/news/2010/10/it-is-possible-\to-kill-the-evercookie.ars.

[36] VEGA, T. Code known as Flash Cookies raises privacy concerns. *New York Times* (September 2010). http://www.nytimes.com/2010/09/21/technology/21cookie.html.

[37] VERNAL, M. An update on Facebook UIDs, October 2010. http://developers.facebook.com/blog/post/422.

[38] ZEIGLER, A. Adobe Flash now supports InPrivate Browsing, February 2010. http://blogs.msdn.com/b/ie/archive/2010/02/11/adobe-flash-now-supports-inprivate-browsing.aspx.