HIDDEN SURVEILLANCE BY WEB SITES

WEB BUGS IN CONTEMPORARY USE*

DAVID M. MARTIN JR.
HAILIN WU
ADIL ALSAID

When a Web browser navigates to a Web page, it usually has to process several distinct network transactions, each referring to a different part of the requested page. These parts include the main HTML source of a Web page and any of its images, scripts, layers, CSS, Java applets, ActiveX objects, etc. Each of these “clickstream” transactions is usually recorded in a remote Web server log file. These log files can be later examined in order to better understand visitor traffic patterns, interests, site problems, and even break-in attempts. Each of these logging and analysis purposes amounts to a form of user surveillance, some more intrusive than others.

Privacy advocates are concerned about the possible uses for these massive data troves, because anyone who knows the URLs that you visit pretty much knows what you are reading on the Web. Given a typical URL, it’s trivial to turn it into the full Web page text – just put the URL into a Web browser. There are currently no federal restrictions on the commercial acquisition and use of clickstream information in the U.S. other than the Federal Trade Commission’s enforcement of site privacy policies under its authority to investigate “unfair and deceptive trade” practices. In other words, the FTC regulates statements about privacy practices, but not the practices themselves. The government does face some restrictions on its own acquisition of clickstream data, but they are low hurdles: the U.S. Patriot Act of October 2001 allows clickstream information to be gathered once the investigators certify before a magistrate that it is “likely […] to be relevant to an ongoing criminal investigation.” The rules appear to be far looser in non-prosecutorial settings. For example, in Knight v. Exeter School District (New Hampshire), the court ruled that a student’s parent had the right to review a school’s squid proxy log [1]. And in Indiana, a reporter cited the state’s open records law in order to obtain browser history, bookmark, and cookie files from 49 school superintendents [2].

Meanwhile, Web site operators tend to downplay the importance of clickstream logs. They point out that server activity logging is routine, indiscriminate, and enabled by default on Web servers. Many site operators promise not to use the data for personal identification purposes unless compelled to under law. One is left with the impression that while the data fills up disk drives, there may be no real plans for it – until a corporate reorganization or a national calamity strikes.

* This research was supported by the Privacy Foundation, the University of Denver, and Boston University.
How then can we distinguish between data collected intentionally for surveillance purposes from data that is collected routinely as part of a Web site’s operation? We approached this question by singling out certain Web elements that appear to have no purpose other than information gathering and that also can also be recognized automatically. Counting these elements then provides a lower – possibly very lower – bound on the amount of intentional tracking in the Web. These elements of interest are Web bugs, which are covert (but not too covert) channels between a Web site and a third party data collector. Web bugs are also known as Pixel Tags, Web Beacons, and Clear GIFs.

We started with two lists of Web sites chosen for inclusion in the U.S. Federal Trade Commission’s 2000 Web privacy study [3]. The “popular” list consisted of 91 Web sites designated as the most popular in January 2000, and the “random” list consisted of 335 consumer oriented Web sites. Some of these sites no longer exist, so we narrowed it down to 84 from the popular list and 298 from the random list. We used a modified version of Bugnosis [10,11], our own Web bug detection engine, to analyze approximately 90 pages from each site on average. We also read the privacy policies of popular sites to see how well they corresponded with the site’s actual practices.

Under our definition, we found that 58% of “popular” Web sites and 36% of “random” Web sites contain Web bugs somewhere close to their home page. That’s a lot! After reading the privacy policies of the popular Web sites, we found that 29% (14 out of 48) of those that do contain Web bugs say nothing about third party content and corresponding surveillance capabilities, let alone the possibility of Web bugs on their sites. Without an appropriate disclosure, the users of these sites really have no way to know that they may be participating in a third party surveillance network. On the brighter side, 67% (26 out of 39) of the third parties we encountered did provide machine-readable (P3P [4]) privacy policies on their sites.

Two other reports have specifically addressed the distribution of Web bugs. Reinke at E-Soft Inc. [5] concentrated on identifying and ranking the third party recipients of Web bug data. His report first ranked the third parties by the number of Web sites that carry their Web bugs. Taking the Web sites’ popularities into account, Reinke also ranked the third parties by the total amount of surveillance data the third parties are likely to receive. The sample space was 701,176 pages retrieved from 101,991 different Web sites.

Murray and Cowart [6] at Cyveillance Inc. considered changes in Web bug practices over time by comparing results from two samples, and concluded that Web bugs are almost 5 times as likely to appear on a random page in 2001 as they were on a random page in 1998. The authors also noted a very strong correlation between Web bug presence and the presence of content from “leading brand names” on Web pages in 2001. Their sample space included pages from over 1,000,000 sites.

1. WHY WOULD ANYONE CARE WHAT WE READ?

In an October 4, 2001 press release announcing the development of standards for Web bug use by its members [7], the Network Advertising Initiative wrote:

“Web beacons are a tool that can be used online to deliver a cookie in a third party context. This allows companies to perform many important tasks – including unique visitor counts, web usage patterns, assessments of the efficacy of ad campaigns, delivery of more relevant offers, and tailoring of website content. The web
According to the Privacy Foundation [8], Web bugs may also be used in order to:

- Count the number of times a particular Web page has been viewed.
- Transfer demographic data (gender, age, zip code, etc.) about visitors of a Web site to an Internet marketing company. This information is typically used for online profiling purposes.
- Transfer personally identifiable information (name, address, phone number, email address, etc.) about visitors of a Web site to an Internet marketing company. This information is typically used for online profiling purposes. It also can be combined with other offline demographic data such as household income, number of family members, type(s) of car(s) owned, mortgage balance, etc.
- Profile individuals by tracking what Web pages they visit across many different Web sites.
- Pass off a person's search strings from a search engine to an Internet marketing company.
- Match a purchase to a banner ad that a person viewed before making the purchase. The Web site that displayed the banner ad is typically given a percentage of the sale.
- Count the number of times a banner ad has appeared.
- Report back the type and configuration of the Internet browser used by a visitor to a Web site. This information is typically used in aggregate form to determine what kind of content can be put on a Web site to be viewed by most visitors.
- Allow a third-party to provide server logging to a Web site that cannot do this function itself.

Keep in mind that while not all of these uses are inherently threatening to privacy, they’re all a form of user population surveillance. Web users are at an inherent disadvantage here, because the surveillance beacon's cookie is typically delivered or read through a single pixel on the host site.”

**WEB BUG EXAMPLES**

**Page URL:** http://fashion.about.com/mbiopage.htm

**Web bug URL:** http://m.doubleclick.net/viewad/390487/pixel.gif

**Associated cookie:** id=8000000d02fc90f

This Web bug placed at about.com is a straightforward example. The URL of the Web bug contains the number 390487, which probably identifies a partnering agreement between about.com and doubleclick.net. The URL itself ends with the name “pixel.gif” – a common name for Web bug images, since they usually contain only a single pixel. Interestingly, the URL also contains the name “viewad”; perhaps this bug is meant to convey the fact that some nearby advertisement was displayed. Finally, the third party cookie associated with the Web bug provider doubleclick.net is a straightforward user ID.

**Page URL:** http://sho.com/queer

**Web bug URL:** http://ad.linksynergy.com/fs-bin/show?id=9ty3z8Cgl5c&bids=28413.10000020

**Associated cookie:** lsn_statp=i%2FOeBA%3D%3D;lsn_track=UmFuZG9tSVbn%2BaRRdAYn4BOQn0uOnzQlAJMMl596jg5ZDcactK%2F7FCH9yylV0OsdnKt7W50uEpM%3D;linkshare_cookie14231=10000013%3A14231

This one was placed on Showtime’s site for “Queer as Folk” – its adult series about the lives and romances of a group of urban gay men. How many people are comfortable with the thought of unseen monitors noting their interest in particular sexual themes?

**Page URL:** http://www.msnbc.com/news/attack_front.asp?0S=B=C944

**Web bug URL:** http://www.qwestdex.com/img/tracking/msnbc.gif?r=473

**Associated cookie:** profile=userid&dmjr&mypwd&0&font&regular&usrfirst&David&userlast&Martin&city&DENVER&state&CO&key1&Home&plat1&39.691399&plng1&-104.928802

This last bug is remarkable for the amount of personal information visible in the cookie. Both a username “dmjr” and password “mypwd” are visible, along with a full name
mechanism is designed to go unnoticed. Think of a hidden video camera at the entrance of a public library. Maybe it’s only there to count the total number of visitors, or maybe it’s recording every face it sees forever. But it hardly makes sense to ask which it is without knowing the camera is there at all.

II. WEB BUGS: A DEFINITION

So what are these Web bugs, really?

**General definition.** A Web bug is any HTML element that (1) is intended to go unnoticed by users, and (2) is present partially for surveillance purposes.

This captures our intent, but we’re going to have real problems writing a computer program that figures out whether an arbitrary HTML element is supposed to be noticed and why it is there. We had to make this definition much tighter.

We start by looking only at third party images that are “tiny” (7 square pixels or less). This takes care of (1) in the general definition. Unfortunately, it’s common practice among Web designers to use tiny images to achieve spacing and alignment effects, since HTML offers no other way to do it. In order to minimize the possibility of confusing spacing images with tracking devices, we narrowed our focus to third party images that have associated third party cookies. Since cookies are the standard mechanism for uniquely identifying users, this step eliminates the third parties who clearly aren’t interested in that type of tracking. And then we exclude any images that appear more than once on the page. This helps because while spacing and alignment images might naturally appear multiple times on a page, there is little sense in embedding more than one identical surveillance device on the same page.

The final ingredient is an “expert knowledge” database of 29 URL patterns [9,10]. Matching a pattern in this database can either force an image to be identified as a Web bug or prevent it from being so identified even if it has the other characteristics of a Web bug. But ultimately, this database was not a big decision factor: only 5% of the Web bugs found in this study are due to a database match, and less than 0.1% of apparent Web bugs were ignored because of a database entry.

**Specific definition.** A Web bug is an HTML element that (1) is an image, (2) is tiny, (3) is served by a third party, (4) has an associated cookie, and (5) appears only once on the page. Or, if the element’s URL matches one of the 29 “expert knowledge” database patterns, then the database decides whether it’s a Web bug or not.

Other researchers [5,6] generally allow more images to be considered Web bugs than we do. For instance, they don’t require a Web bug to have an associated cookie or to appear only once on the page. But they’re more demanding about the image size – while we’ll consider a 7x1 image small enough to be intended to go unnoticed, they consider only a 1x1 image – the tiniest possible image – to be hidden.
We used Bugnosis [10] to collect and partially analyze the Web pages we captured. Bugnosis is a Web bug detector that we built as a simple privacy monitor add-on for Internet Explorer. Over 100,000 users have installed it; see [11] for more details. One of the purposes of this study was to improve the Web bug detection rules for the next version of Bugnosis, so the definition we used in this study is somewhat different than that used by the current version. Figures 1 and 2 show how Bugnosis shines light on the Web bugs it finds.

![Figure 1: A page as intended to be seen.](image1)

![Figure 2: The same page as modified by Bugnosis, which turned the originally invisible 1x1 images from ad.doubleclick.net in the left side search box into orange bug images. (The eyeball in the bottom status line warns the user that Internet Explorer blocked some content that was inconsistent with its privacy settings – in this case it was a cookie unrelated to this Web bug.)](image2)

III. CONCLUSION

The fact that over 1/2 of random sites and over 1/3 of popular sites use Web bugs means that even more sites are doing some type of surveillance, since Web bugs are really only one approach to generating log entries. This points to an extremely significant level of routine surreptitious surveillance in the Web, and ultimately, to the inappropriateness of the popular understanding of the Web as an electronic library. Libraries and bookstores have no effective means to track the idle browsing of their offerings, but Web sites do, and they use it. When a library visitor presents a library card and checks out a book, they can be confident that the librarian’s record keeping will
be limited to making sure the book is eventually returned, and perhaps noting how popular the book is. That’s just how libraries work in the United States. Even bookstores tend to fight when faced with a subpoena for purchase records. But visiting a Web site, users really don’t know whether they are contributing to a data bank about the site, or about themselves, or both. Every Web site could be different. Actually, users probably don’t even think of the data bank when surfing.

We are left with some questions. Why do Web designers seem to feel so differently about user tracking than librarians do? Is this Web surveillance regimen more due to technical ease or to business necessity? And if surveillance is a given, how can we at least improve user awareness? P3P [4] begins to help, by relieving users of the absurd requirement that they track down all of the constituent pieces on the Web pages they visit and read each privacy policy before deciding whether to proceed. But P3P is essentially an automatic translator, and translators never get to choose the underlying message. Today, the message is that commercial Web sites want automated feedback, and that mainstream users should not worry about generating a remote clickstream log that could be analyzed, searched, archived, and subpoenaed.

IV. REFERENCES

1 Knight v. Exeter School District, Rockingham County Superior Court (New Hampshire), November 2000.
4 Platform for Privacy Preferences Project (P3P), World Wide Web Consortium, http://www.w3c.org/P3P.
11 Adil Alsaid and David Martin, Detecting Web Bugs With Bugnosis: Privacy Advocacy Through Education, 2001, submitted. (This citation will soon point either to conference proceedings or a tech report; for now the paper is available at http://www.cs.bu.edu/~dm/pubs/bugnosis-pet2002.pdf.)