Priv3: A Third Party Cookie Policy

Mohan Dhawan
Rutgers University
mdhawan@cs.rutgers.edu

Christian Kreibich
ICSI & UC San Diego
christian@icir.org

Nicholas Weaver
ICSI & UC San Diego
nweaver@icir.org

ABSTRACT

In today's World Wide Web, there exists significant economic pressure to track user activity, a development users may find objectionable. Safari's third-party cookie policy works well to block tracking from advertisers and other pure third-party content, but is insufficient to block the multi-function tracking present in *Third-party* widgets such as "Like" buttons and other "social plugins" offered by the likes of Facebook, Google, and Twitter. These elements provide desired user functionality, but also expose the user to the possibility of cookie/referrer-based *tracking* by those third-party sites. Naïve approaches that completely disable third-party interactions prevent such tracking, but at the same time break desired tasks, such as "liking" a page, or engaging in a discussion forum. In order to enable a middle ground, we present Priv3, a web browser extension which uses conditional suppression of third-party cookies and selective reloading of elements on a web page to provide a generic mechanism to protect user privacy from these trackers without compromising usability, creating an "allow with user intent" third-party cookie policy. We have made Priv3 available as Firefox extension that has been downloaded 97,000 times to date, featuring an average user base of 17,000 users daily.

1 Introduction

We believe that any "Do Not Track" mechanism which relies solely on voluntary compliance will not work. Thus absent the force of law, we need to develop technical mechanisms which prevent tracking while still enabling functionality. The Safari third party cookie policy, which we describe as "allow due to previous interaction", works to block tracking from advertisers, but it can't block the tracking performed by various social widgets as when a user logs into one of the social sites, Safari now allows the site's cookies as valid third-party cookies.

By design, these widgets also allow their providers—often companies that specialize in advertising—to track user activity, leading one to wonder as to the extent to which these widgets were designed to track and profile users.

Classic examples of such dual-purpose trackers include Facebook's "Like" button and the "Comment" box. These elements both provide information to users (the number of "likes" and the comments on the article) and enable mechanisms for the users to interact with these elements. However, they also conveniently notify Facebook that a particular, logged-in Facebook user is currently viewing the page, a circumstance valuable for crafting targeted advertisements. Not infrequently, users remain ignorant of the fact that this *third-party privacy leakage* is taking place.

Unlike pure third-party tracking as employed e.g. by advertising networks, simply blocking third-party cookies disrupts desired functionality: users find themselves unable to "Like" pages or enter comments. Such blanket prohibitions may even break some of the functionality provided by the web sites themselves. The central bit of information that allows third parties to associate the rendering of a widget with a particular user is the HTTP cookie that the third party plants in the user's browser upon first contact. Preventing the cookie information from reaching the third party prevents (or at the very least substantially complicates) the association with the user. Accordingly, recent solutions to the problem [3, 15] disable third-party elements completely, replacing or removing them until a user performs an explicit action. Such user intervention to enable third-party cookies is annoying and hampers usability.

We present Priv3, a browser extension which uses a generic mechanism of conditional suppression of third-party cookies and automatic reloading of selected web page components to protect user privacy from both social and web trackers, without compromising usability, which we describe as "allow with user intent". Priv3 takes advantage of these social features also supporting anonymous (non-logged-in) users, by initially loading all the elements associated with Google, Facebook, Twitter, and LinkedIn without cookies. Thus even without cookies, these widgets will still display the number of "likes," comments, and other features. When a user then chooses to interact with a third-party widget visible on the page, such as entering a comment using Facebook's "Comment" box, Priv3 detects user intention to interact with the page and automatically reloads with cookies only the selected components on the web page belonging to the third party. The entire process remains unobtrusive and does not interfere with the user's overall web browsing.

We demonstrate that conditional suppression of third-party cookies and automatic reloading of selected elements on the web page provides a generic defense against both social and web trackers, and that we can achieve the desired functionality without hampering end-user experience. We have built and released Priv3 as Firefox browser extension on AMO [1]. To date, users have downloaded Priv3 97,000 times and we have an average user base of about 17,000 users daily.

The rest of the paper proceeds as follows. Section 2 discusses web tracking and the existing defenses against it. In Section 3, we

present details about the design of Priv3. In Section 4, we discuss our experience with Priv3and finally conclude in Section 5.

2 Overview

2.1 Background

Most web sites include scripts and widgets from third-party sources for the purpose of providing useful services, such as personalization or social interaction. These elements generally run within <iframe> instances, allowing them to set and access cookies within the third party source, with information about the page containing the <iframe> passed within the URL of the <iframe> itself.

For example, a site which wishes to include the "Like" button in a web page either directly creates an <iframe> pointing to a Like button URL on Facebook's site or includes a script and a HTML which the script replaces with the desired <iframe>. In either case, the URL within the <iframe> includes the URL of the desired page, and since the <iframe>'s domain is `facebook.com`, any user cookies are also transmitted. Thus Facebook sees that the identified user visited the specific page. Such third-party elements need to render within their own <iframe> instances in order to properly transmit the social site's session cookies.

Note that while these elements work to track users by their session cookies, they can still provide useful information to users in the absence of session cookies, i.e., when the user has not logged on. The "Like" button, for example, still renders meaningfully, displaying the total number of "Likes." The Facebook comment plugin still displays the comments on the page.

Unlike the tracking employed by advertising networks and analytics tools, these trackers are *multi-functional*: the designer of the widget-including site would like the visitors to gain some direct benefit. Thus a policy which simply blocks third-party cookies, or the widgets altogether, would prevent users from adding comments, "liking" pages, or performing other relevant activities.

2.2 Current defenses

The problem of privacy violation through web tracking is well documented in prior work [6, 8, 10, 15]. Several solutions have been proposed by browser vendors and researchers alike to ensure user privacy while browsing. Based on their implementations on the client, we categorize these approaches as follows:

Safari's Third Party Cookie Policy. Most web browsers allow preference settings for clients to suppress acceptance of third-party cookies. Safari uses a unique default policy for these third-party cookies, which may be described as "allow due to previous interaction". Only if the user has previously interacted with the site are the third-party cookies allowed. This protects the user from tracking by pure advertising or analytics sites, as these sites are unable to set or read third party cookies. Although effective at blocking advertising-related tracking, this can't protect the user from tracking by social widgets.

Do Not Track. Modern browsers also implement the Do-Not-Track proposal [9]. When this header is present in the HTTP requests, it signals a user's intent to opt out of third-party web tracking. While Do-Not-Track provides users with a simple mechanism to safeguard their privacy, it is not binding on the third-parties to honor the directive and requires strict regulation to ensure compliance.

Extensions. Due to the limited customization available in the built-in functionality provided by the web browsers, developers often use browsers' APIs to build privacy-enhancing extensions. Most of these extensions require user input to enable a flexible cookie management policy, i.e., either a complete or universal cookie blocking or even selective cookie suppression for specific domains. A major problem with the above approach is its inflexibility: once the browser has loaded the page, it will not relax its cookie policies dynamically.

To overcome this issue, another line of extensions, like Ghostery [3] and ShareMeNot [14], explicitly request the user to reload the web page with selected third-party cookies enabled.

- **Ghostery** is a browser extension which provides complete blocking of a large number of third-party trackers and advertisers. Recent versions of the extension also attempt at blocking social tracking by removing all social widgets by default. The user must explicitly click on additional Ghostery introduced buttons to reload the page with widgets enabled. Although effective, this not only affects site layout and introduces a significant page change when an element is enabled, it also prevents the anonymous viewing of comments and similar features.

- **ShareMeNot** is a privacy extension specifically designed for protecting users from tracking by social networks. ShareMeNot replaces the social network buttons with ShareMeNot's internal buttons which, when clicked, pass the click to the social network. This approach changes the page layout and cannot process any elements that the extension is not specifically configured to support. ShareMeNot does not support the Facebook Comments widget, for example, so the extension simply removes those elements from the page without providing a mechanism to see the comments anonymously and browse the content.

HTML and JavaScript. Both <iframe>s and the same-origin policy are useful but they by themselves are not sufficient to stop web or social trackers. Recent advances in web technology allow content publishers, i.e., hosting web sites, to limit transmission of information. HTML5 proposes a new "noreferrer" attribute value [4] which directs the browser to remove `Referer` headers from the specified HTTP request, while the recently introduced Content Security Policy (CSP) [11] by Mozilla can be used to suppress web tracking by specifying a list of pre-approved domains available for communication.

3 Design and Implementation

Priv3 is a browser extension designed for protecting user privacy against web trackers. As mentioned in Section 2.2, prior work provides limited usability, and such extensions often require user intervention or introduce rendering artifacts. Thus, a key idea we incorporated into Priv3 was to observe the user's intent and enable automatic reloading of cookies. We also desired that our design be generic: while we target specific sites (Google, Facebook, Twitter, and LinkedIn), our tool does not need to understand the implementation of specific elements. Finally, we desired that our tool be both transparent and, to as large a degree possible, unnoticeable. Thus we introduce no changes within the page design, and any content refreshes only represent the transition from an logged-out to a logged-in state on any social widgets. We describe this third-party cookie policy as "Allow with user intent".

User intent. We define user intent as an explicit action on behalf of the user to interact with visible elements on the web page and allowing release of any personally identifiable information associated with it. For example, we would like that unless the user explicitly clicks the "Like" button to record his preference, no third party would receive the cookies, but that once a user expresses an

intent to Like something, the button should work normally. We infer user intent by keystroke and mouseclicks. If the user directs a keystroke or a mouseclick to a social element, we believe it is clear that the user wishes to interact with the element, making it safe to fully enable.

When a user visits a web page on a browser enhanced with Priv3, all the targeted multi-purpose trackers are loaded without cookies, causing them to behave as if the user is not logged in. During a browsing session if the user expresses intent to interact with a particular third-party widget, Priv3 reloads all the DOM elements on the web page belonging to the third-party site with access to the domain cookies.

This selective cookie suppression and reloading of parts of the web page ensures that only trusted components on the web page have access to the user's personal information. This mechanism is seamless and unobtrusive thereby user-friendly.

Implementation. We implemented Priv3 for the Firefox web browser and it is available on Mozilla's addon gallery.[1] We now discuss a few of the salient issues in the implementation.

(1) Capture intent. Priv3 intercepts user mouse clicks and key strokes to identify the target of the event, usually a visible component, like a hyperlink. But in several cases the hyperlink might itself be embedded within an `<iframe>` from another domain. Priv3 uses Firefox's APIs to precisely identify the exact event target and later uses this information to selectively reload third-party components.

(2) Access control. By default, when a web page loads, Priv3 removes the cookie headers in the HTTP requests to third-party domains. This ensures that if the user does not wish to interact with the third-party then it does not learn the identity of the user. But, simply scrubbing out the HTTP cookie header does not prevent the remote server from not learning the identity of the user.

A third-party JavaScript script code can invoke `document.cookie` to gain access to the domain cookies, which could later be transferred to the third-party by circumventing the same-origin policy. To disable such accesses, Priv3 uses script execution event handlers [12, 13] which are fired just before and after the browser executes the corresponding JavaScript code. These handlers control access to the browser's cookie store, i.e., before the third-party script is to be executed, the handler disables access to the cookie store to all JavaScript and later restores the permission after the script has finished execution. Thus, all third-party JavaScript code executing on the web page has no access to either the third-party cookies or the domain cookies until the user chooses to interact with it.

(3) Selective reload. Once the user's intent to interact with a third-party has been established, Priv3 reloads all components from the intended third-party domain. If the target of the user intent was within a third-party `<iframe>`, Priv3 reloads the `<iframe>` with the domain cookies enabled. In certain cases, a click on a hyperlink opens up a new popup or window pointing to a third-party domain. Priv3 identifies such user intent to navigate to a third-party site and ensures that the ensuing HTTP request to the third party has access to the domain cookies. It also reloads JavaScript code from a third-party domain with access to domain cookies.

4 Discussion

Priv3 is a relatively small Firefox extension, requiring less than 700 lines of code. This small size complements the simple nature of the selective-reloading mechanism. Although we currently only block the four major social trackers, it would be trivial to extend to other domains. In particular, the extension does not contain code to recognize individual social elements, only the domains which host social elements. We initially released Priv3 a year ago, with largely positive feedback from our users.

It also remains transparent to the user, as we do not replace the implementation of buttons or social elements. The social networks themselves ensure that the logged-in and logged-out states share similar dimensions, thus the page layout does not deviate from the site's intent. When the user authorizes the social element, only the `<iframe>` instances themselves reload, preventing any disruptive visual effect. We find the clearest indication of the effect manifests on Facebook comment elements: when the user clicks the mouse to input text, the *only* noticeable change on the page consists of the user avatar image changing from the default logo to the user's profile photograph.

We believe this transparency matches user expectations. We believe that users generally do not expect the "Like" button to report to Facebook that a user views a page, but do expect that a *click* on the "Like" button does. Our goal was to make the behavior match this expectation: only when the user "likes" something should Facebook learn about it.

Cookie access control. Priv3 uses access control over cookies to perform its functionality. Its access control utilizes a blacklist of domains to prevent transfer of third-party cookies. Thus, the blacklist must be updated every time the third party introduces resources fetched from a new affiliate domain. This proves essential for preventing the third party from subverting Priv3 by storing tracking meta-data in the cookies from the new domain. Fortunately this is a very coarse-grained blacklist, as it does not need to understand new social elements. For example, to block access for Google+, Priv3 blacklists the following domains: `google` `gstatic`, `google-analytics` and `youtube`.

Priv3 does not enable fine-grained cookie management for end users or modify any existing user-defined cookie policies (such as "allow-for-session"), or introduce any new rules. It simply prevents JavaScript fetched from third-party sources from accessing the hosting web site's domain cookies, until the user expresses his intent.

Aggregators. A few third-party widgets, including AddThis [2] or ShareThis [5], provide access to a number of social networks. These aggregators are themselves web trackers which in turn facilitate social tracking. Priv3 supports user intent based cookie blocking mechanism for such aggregators as well. If a user visits a page with the aggregator widget, Priv3 blocks the aggregator's cookies unless the user specifically visits or logs into the aggregator's website. Once the user interacts with a social network of his choice as displayed on the aggregator widget, cookies corresponding to the social network are enabled across the web page. Thus, Priv3 proves effective against both social and web trackers.

Evasion. It would be possible for a social network to evade our current implementation. Because we still enable loading, albeit without cookies, passive tracking techniques [7] and IP-based tracking may still identify users' history. Similarly, our technique does not currently block HTML5 or flash local storage. However, actually exploiting these weaknesses would be politically dangerous for the social networks, as they would be actively subverting user privacy expectations, and the technique could be extended to ensure that the `<iframe>` instances do not have access to local storage, with the browser string replaced with a known common string.

[1] Priv3 can be downloaded from `https://addons.mozilla.org/en-US/firefox/addon/priv3/`.

5 Conclusion

We have presented Priv3, a browser extension which uses conditional suppression of third-party cookies and automatic reloading of selected web page components to provide a generic defense against both social and web trackers. Social widgets can only track a user by cookie when the user actually interacts with the element, instead of simply viewing the web page containing the element.

We show that this "allow on user intent" cookie policy enables Do Not Track functionality using purely technical means which applies not only to advertisers (which are blocked by Safari's "allow on previous interaction" policy) but also social widgets which serve to both track users and provide functionality. This policy protects users from unwanted tracking without requiring either voluntary compliance or the force of law.

Priv3 accomplishes the desired functionality without hampering end-user experience, as the social elements all remain in the page with no cosmetic change, apart from the transition from the unlogged-in to the logged-in state when the user interacts with an element.

We have built and released Priv3 as Firefox browser extension on Mozilla's addon gallery. It has been downloaded 97,000 times to date, and has an average user base of about 17,000 users daily.

6 References

[1] Addons mozilla. https://addons.mozilla.org.

[2] Addthis. http://www.addthis.com/.

[3] Ghostery. http://www.ghostery.com/.

[4] HTML5 noreferrer. http://www.whatwg.org/specs/web-apps/current-work/multipage/links.html#link-type-noreferrer.

[5] Sharethis. http://sharethis.com/.

[6] Berkeley Law, UC Berkeley. Web Privacy Census. http://www.law.berkeley.edu/privacycensus.htm.

[7] EFF. Panopticlick. https://panopticlick.eff.org/.

[8] Dongseok Jang, Ranjit Jhala, Sorin Lerner, and Hovav Shacham. An empirical study of privacy-violating information flows in javascript web applications. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 270–283, New York, NY, USA, 2010. ACM.

[9] Jonathan Mayer. Do Not Track. http://tools.ietf.org/id/draft-mayer-do-not-track-00.txt.

[10] Jonathan R. Mayer and John C. Mitchell. Third-party web tracking: Policy and technology. In *IEEE Symposium on Security and Privacy*, pages 413–427, 2012.

[11] Mozilla. Content Security Policy. https://dvcs.w3.org/hg/content-security-policy/raw-file/tip/csp-specification.dev.html.

[12] Mozilla. onafterscriptexecute. https://developer.mozilla.org/en/DOM/element.onafterscriptexecute.

[13] Mozilla. onbeforescriptexecute. https://developer.mozilla.org/en/DOM/element.onbeforescriptexecute.

[14] Franzi Roesner. Sharemenot. https://addons.mozilla.org/en-US/firefox/addon/sharemenot/.

[15] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. Detecting and defending against third-party tracking on the web. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, NSDI'12, pages 12–12, Berkeley, CA, USA, 2012. USENIX Association.