The hidden perils of cookie syncing

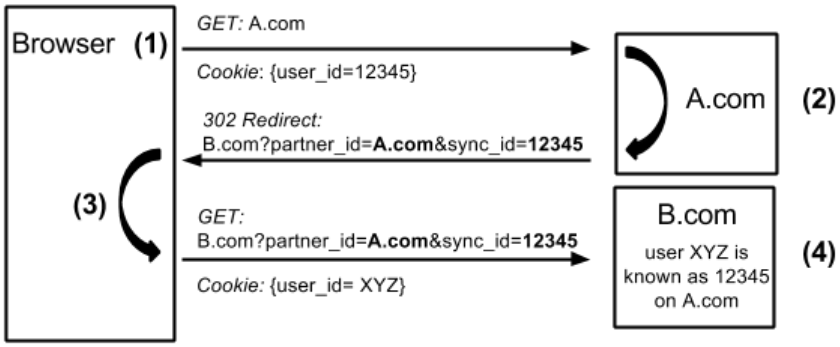
AUGUST 7, 2014 BY STEVEN ENGLEHARDT

*[**Steven Englehardt** is a first-year Ph.D. student in the computer security group at Princeton. In this post he talks about the implications of a recent study that we published in collaboration with researchers at KU Leuven, Belgium. — Arvind Narayanan]*

Online tracking is becoming more sophisticated and thus increasingly difficult to block. Modern browsers expose many surfaces that enable users to be uniquely identified, including Flash cookies and browser fingerprints. In a **new paper** that will appear at **ACM CCS**, we present the first large scale study of three advanced tracking mechanisms — **canvas fingerprinting**, **evercookies**, and **cookie syncing**. We developed novel measurement techniques and found that these tracking mechanisms are used on a large number of sites. Our findings on canvas fingerprinting, in particular, have been in the news (**Propublica**, **BBC**, **EFF**).

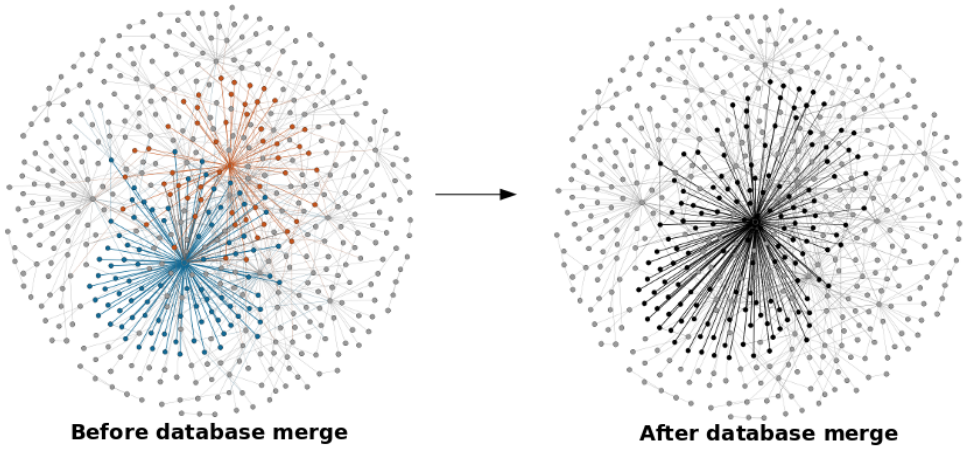
In this blog post I'll focus on a different part of our paper that looked at *cookie syncing*, the process by which two different trackers link the IDs they've given to the same user. The most common use of cookie syncing is to enable **real-time bidding** between several entities in an ad auction. It allows the bidder and the ad network to refer to the user by the same ID so that the bidder can place bids on a particular user in current and future auctions. **Cookie syncing raises subtle yet serious privacy concerns,** but due to the technical complexity of explaining it, didn't receive much press coverage. In this post I'll explain cookie syncing and why it's worrisome — even more so than canvas fingerprinting.

In our study, we measured the prevalence of cookie syncing using our newly-built web measurement platform, **OpenWPM**. The platform allows us to automate visits to a site and record all HTTP traffic and changes to the browser's state that result from the visit. We can use this data to trace the flow of third-party cookies that contain unique identifiers[1]. We found that nearly 40% of all tracking IDs are synced between at least two entities, so it is a ubiquitous practice.



How cookie syncing works. The process begins when a user visits a site (say *example.com*, not shown in the figure), which includes *A.com* as an embedded third-party tracker. **(1)** The browser makes a request to *A.com*, and included in this request is the tracking cookie set by *A.com*. **(2)** *A.com* retrieves its tracking ID from the cookie, and redirects the browser to *B.com*, encoding the tracking ID into the URL. **(3)** The browser then makes a request to *B.com*, which includes the full URL *A.com* redirected to as well as *B.com*'s tracking cookie. **(4)** *B.com* can then link its ID for the user to *A.com*'s ID for the user[2]. All of this is invisible to the user.

Once two trackers sync cookies, they can exchange user data between their servers. This data can be browsing histories or even PII[3]. This exchange doesn't go through the browser, and so it cannot be observed by experiments like those in our study. To be clear, we don't know if this is a common practice. But this is precisely my point: cookie syncing enables a world of back-end data sharing, and there is so little oversight of the tracking ecosystem that we just don't know what is happening behind the scenes. And this is a problem. Based on the evidence of what we *can* observe in the browser, it seems that every avenue for data collection and sharing does seem to eventually get utilized.

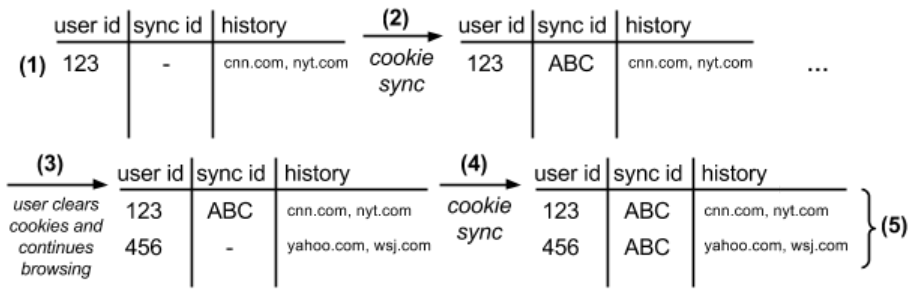


(**Click to play**)

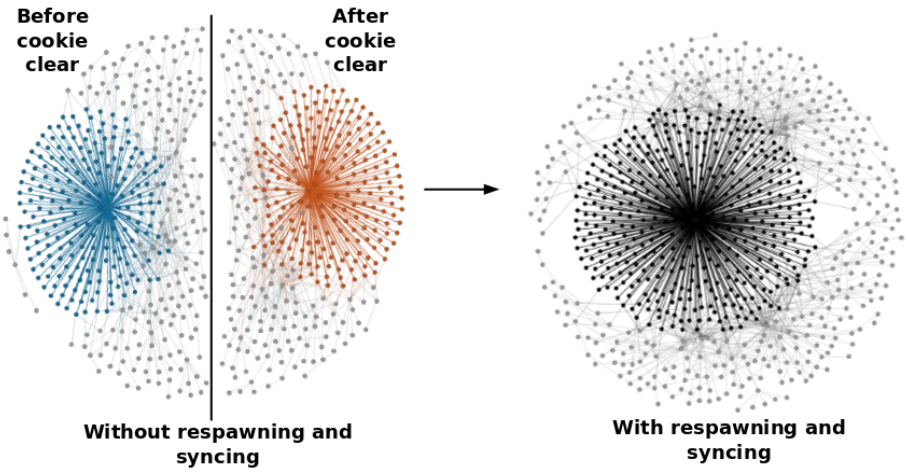
The graph shows what happens when a user visits several hundred of the **top 1500 Alexa sites**. Each node is either a site visited by the user or a tracker; each site is connected to each of the trackers that it embeds[4]. Two specific trackers are highlighted in blue and orange, along with the first-party sites that they track. Clicking the graph will display an animated version which shows how two trackers could merge tracking history after sharing tracking IDs. Essentially, it is as if there is a single tracker with the combined reach of the two.

To put it concretely, in our measurements we found only two trackers (doubleclick.net and googleanalytics.com) that are present on 40% or more of websites. But if we assumed a moderate amount of back-end data sharing (defined in Section 5.3 of **our paper**), the number of trackers that can observe 40% of users' browsing history would jump to 161.

Cookie syncing can also negate the effect of users clearing cookies. How might this happen? Some trackers "respawn" cleared cookies, typically by abusing browser features. This is called an "evercookie." (We studied and measured evercookies in a separate section of **the paper**.) Fortunately, the practice is considered an egregious privacy breach and none of the major tracking companies do it. But here's the kicker: we found some trackers respawning their cookies after the user clears all cookies, and passing these respawned cookies to other trackers via cookie syncing! **Thus, even trackers that don't employ respawning/evercookies nonetheless gain the ability to continually track users who clear cookies**, as we explain below.



The diagram shows the records stored in tracker.com's databases as a user browses various sites: **(1)** tracker.com tracks the user with cookie ID *123*. **(2)** tracker.com receives the synced ID *ABC* from partner.com. **(3)** The user clears her cookies and tracker.com begins tracking the user with a new cookie ID *456*. However partner.com respawns the same ID of *ABC* (not shown). Without external information, tracker.com is unable to link IDs *123* and *456* to the same user. **(4)** partner.com syncs the value *ABC* with tracker.com **(5)** tracker.com knows *123* and *456* correspond to the same user since they are linked with partner.com's cookie ID *ABC*. Now tracker.com has the ability to link the user's histories under IDs 123 and 456, if it chooses to do so.



- **2005:** J F M A M J J A S O N D
- **2004:** J F M A M J J A S O N D
- **2003:** J F M A M J J A S O N D
- **2002:** J F M A M J J A S O N D

author log in

(**Click to play**)

This graph shows what happens when a user visits several hundred of the top 1500 sites in a random order but clears cookies midway[5]. As before, nodes represent pages visited or embedded third-party trackers and edges represent a tracked visit. We highlight the same tracker both before and after the user clears cookies (blue and orange respectively). The graph on the left is completely disconnected, meaning the user effectively appears as two different users to the trackers. But when this tracker receives respawned cookies via cookie syncing, they gain the ability to connect the user's visits to websites they track from before and after cookie clearing.

In our study, we found a relatively obscure tracker *merchenta.com* respawn an ID on a single site out of 3,000 we crawled. It then passed this ID to *adnxs.com*, a major tracker which we observed on approximately 11% of the sites but does not respawn cookies itself. Again, to be clear, we have no way of knowing if adnxs.com links browsing history in this manner as a matter of policy, but the information needed to do so resides in their databases and that is a problem by itself.

In summary, cookie syncing can enable trackers to create more complete and persistent profiles on the users they track. Given the sophistication of today's trackers, **starting a truly fresh browsing profile is a very difficult task — the web never really forgets**.

Let me end by pointing out why cookie syncing is more troublesome than most other tracking techniques including fingerprinting. In response to our work, AddThis, who had by far the most widely deployed canvas fingerprinting code in our study, has stated that they **are ceasing the practice** of fingerprinting. They are able to do this since fingerprinting is not essential to the web or to their business model. On the other hand, cookie syncing is essential to ad auctions; despite its privacy risks, it's not realistic to expect that it will go away. So what's the way forward?

Technical solutions like cookie **double-keying** or **list-based** and **heuristic-based** blocking tools can help prevent cookie syncing to the extent they prevent tracking altogether. However, the business model of the majority of the web will likely prevent the widespread deployment of these solutions as on-by-default for the average consumer. Instead, these consumers must simply trust that companies are not misusing tracking data. Transparency of data use practices in an externally verifiable manner would go a long way toward repairing consumer mistrust of online tracking and advertising.

[1] ID cookies are determined using the method described in Section 5.1 of our paper.

[2] Alternative syncing mechanisms exist; A.com can build the request to B.com in javascript (requiring a single request), or B.com can generate the initial request to A.com with its tracking ID embedded in the url (thus performing a two-directional sync).

[3] Onboarding companies are an example of trackers that receive PII from other domains.

[4] This graph shows real data collected during a measurement, where the front page of the global top 3000 Alexa sites were visited in descending order. Nodes are Public-Suffix + 1, edges are the existence of an HTTP referrer back to the visited site and an identifying cookie. For graph simplicity Facebook.com is excluded and only a subset of page visits are included.

[5] We use the same data as before, but simulate this by splitting tracker nodes into pre-clear and post-clear.

Written with many excellent contributions from Christian Eubank and Arvind Narayanan. Thanks to Joseph Bonneau and Ed Felten for their helpful comments.

FILED UNDER: PRIVACY & SECURITY TAGGED WITH: COOKIE SYNCING, ONLINE TRACKING, PRIVACY, WPM

Comments

dr2chase says:
August 7, 2014 at 9:51 am

Having a good time with Little Snitch and a filter set to bug me about requests to visit adnxs.com, doubleclick.net, merchenta.com and voxmedia.com. Blocking them doesn't seem to have harmed my browsing experience.

Mehdi says:
August 7, 2014 at 4:29 pm

Hi very interesting research and results. Is the HTML5 storage (local, session) where the cookies are stored or is it something else/more ? I've been looking for some info there with "firestorage" and the quantity and depth of information stored there is pretty impressive. Have you been exploring it ?

Also I installed OpenWM seamlessly (great work btw) and now I have a database that I would like to read. How could I do that ?

Steven Englehardt says:
August 12, 2014 at 12:44 pm

Thanks! We did look into HTML5 storage, but the analysis presented here soley looks at standard HTTP cookies.

I recommend the Firefox add-on: https://addons.mozilla.org/en-US/firefox/addon/sqlite-manager/ or SQLite Studio for database exploration.

Wes says:
August 7, 2014 at 5:56 pm

FYI, doubleclick.net and googleanalytics.com are both Google: http://en.wikipedia.org/wiki/DoubleClick

Knowing this you can guarantee that they are sharing data.

Menachem was here says:
August 7, 2014 at 6:57 pm

I use dozens of different devices (not exaggerating) to surf the web. I'd be very surprized if They've linked all of them back to me, whoever I am.

Boris Yeltsin says:
August 7, 2014 at 6:58 pm

The other technique is to only visit webpages that you're not actually interested in.

Valerie O'Neill says:
August 8, 2014 at 2:14 pm

There needs to be a universally recognised signal i.e. DNT:1 (or even better the absence of DNT:0), backed by effectively enforced data protection and privacy law. Luckily we already have the signal (DNT:1 is implemented in virtually all browsers), the law is well established in Europe, and will soon have teeth.

Anonymous says:
August 8, 2014 at 4:54 pm

These results are nice, but not really new:
http://www.internetsociety.org/doc/selling-privacy-auction

Mitch Golden says:
August 11, 2014 at 6:47 pm

Self destructing cookies plugin goes a long way to defeating this.

https://addons.mozilla.org/en-US/firefox/addon/self-destructing-cookies/

Of course, disallowing third-party cookies altogether is also a good idea.

kalakal says:
August 11, 2014 at 7:21 pm

I'm confused as to how b.com knows the identity of the user if the user clears all cookies. I'd understand cookie syncing working if the user only cleared cookies from a.com, but if cookies from both a.com and b.com are cleared simultaneously, how does b.com still know the identity of the user?

Steven Englehardt says:
August 12, 2014 at 12:57 pm

B.com can still know the identity of the user by making use of cookie respawning (see: http://ashkansoltani.org/2011/08/11/respawn-redux-flash-cookies/ for a technical overview), or by some type of fingerprinting (see: http://en.wikipedia.org/wiki/Device_fingerprint). This allows B.com to re-link user history pre-clear and post-clear. If they then cookie sync that same ID, it could allow partners (A.com) to do the same re-linking, even though those partners don't do any respawning/fingerprinting themselves.

kalakal says:
August 12, 2014 at 6:07 pm

So they use a super-cookie or browser fingerprinting presumably using a combination of IP, user agent string, a list of addons you're using, your supported fonts, ect…

Return to top of page