The best step you can take is to [employ private browsing modes](http://www.macworld.com/article/3152125/security/how-to-browse-privately-and-avoid-persistent-tracking-on-a-mac-iphone-and-ipad.html) in browsers that only store all sorts of locally cached data while a private session is active. As soon as you close all incognito windows or exit a private mode, all sorts of cookies get dumped along with the cache. In fact, the coiner of the term “evercookie,” Samy Kamkar, suggested when he [released his proof of concept in 2010](http://samy.pl/evercookie/) that Safari’s private browsing mode prevented all evercookie methods from persisting at that time!

**A cookie made of iron**

The evercookie in principle, which varies in implementation, makes use of JavaScript to *respawn*: whenever it finds that a given nook into which it’s tucked a unique ID for your browser has had its marker deleted, it creates it again. Whenever you visit sites that make use of these techniques, they not only try to find a lingering ID but also spread them again in case you delete a cache or find some other way of defeating them.

Not every site that uses persistent respawning cookie technology makes use of every method to hide its network IDs, however. The more you scour, the more likely you are to remain less tracked or not tracked at all.

The best step you can take is to [employ private browsing modes](http://www.macworld.com/article/3152125/security/how-to-browse-privately-and-avoid-persistent-tracking-on-a-mac-iphone-and-ipad.html) in browsers that only store all sorts of locally cached data while a private session is active. As soon as you close all incognito windows or exit a private mode, all sorts of cookies get dumped along with the cache. In fact, the coiner of the term “evercookie,” Samy Kamkar, suggested when he [released his proof of concept in 2010](http://samy.pl/evercookie/) that Safari’s private browsing mode prevented *all* evercookie methods from persisting at that time!

But it’s not convenient to always use private browsing, because some of what you want to do requires persistent logins. A mix of private and regular windows may help, but you’re likely spend at least a good portion of your time using “public” browser.

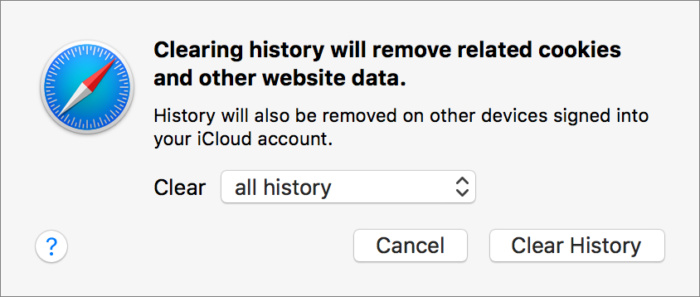
Since Kamkar released his discovery of the many paths by which evercookies can be deposited, some paths to leave traces have been closed down or are easier to disable, while others are even harder.

At the most basic level, with above-board browser cookies, you can control how they’re retained and delete them in a straightforward way. I wrote earlier this year about [disabling third-party cookies](http://www.macworld.com/article/3106793/security/disable-third-party-cookies-as-a-hedge-against-a-new-browser-based-attack.html), which are often connected to ad networks. You can also drop into various places depending on browser and sweep cookies out, deleting them individually by site or name, or dumping all stored cookies. This helps with the most reputable ad technologies.

Evercookies often rely on the two major streaming video browser plug-ins, Microsoft Silverlight and Adobe Flash. These plug-ins allow their own caching and storage, which can be used across sessions and even across browsers. Silverlight is still in use, and remains a requirement for some streaming video services, but you can typically uninstall it if you can’t find a current use.

[The decrepit and insecure Flash](http://www.macworld.com/article/2993361/security/how-to-uninstall-flash-player-from-your-mac.html) fell out of favor long ago, and its use is deprecated by browser makers. In macOS Sierra, [Safari doesn’t load Flash by default](http://www.macworld.com/article/3083841/security/safari-10-to-turn-off-flash-by-default.html) and doesn’t even tell a Web server what kinds of multimedia technology it can display. If a site still tries to push Flash, you have to enable it with a click for one-time or continued use.

Several other methods rely on browser caches and HTML5 storage mechanisms, which were designed to let Web apps more easily store information in a browser for the browser to act upon locally and even without an Internet connection.

Glenn Fleishman

Safari for macOS lets you wipe history and other data, but you also need to use a developer option to dump all caches.

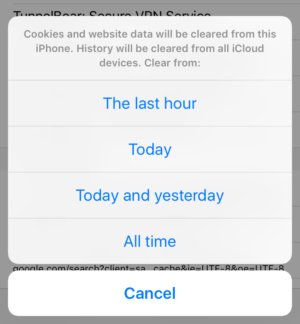
Not all these elements can be deleted by a user, although routinely deleting your caches can help. In macOS with Safari, you’ve got two options: light and nuclear. For the “light” option, in Safari choose Safari > Clear History and select All History from the popup menu, and [a huge number of items get wiped out](https://support.apple.com/kb/PH21412?locale=en_US), including visited pages and browser cookies. Wiping history deletes it across all linked iCloud accounts, too.

The nuclear option requires enabling the Develop menu in Safari > Preferences > Advanced. Check Show Develop Menu in Menu Bar. From the Develop menu, choose Empty Caches. There’s no warning or going back once you select the item. This deletes all local storage, all browser cookies, and everything cached relatived to the Web browser.

Close all tabs, nuking the caches, erasing history, quitting the browser, and relaunching revealed no evercookies hiding. But now I have to log back into every site I use routinely. With 1Password, that’s not hard, but it’s definitely a price you pay. And any offline storage usage by legitimate sites has been wiped out, too.

In iOS, you have two methods to wipe all Safari data when you’re not using Private Browsing Mode:

* In Safari, tap the Bookmarks button, then tap the Bookmarks tab, and tap History. Tap the Clear button at the bottom, and you can select a duration; tap All Time.
* Alternatively, you can use the Settings app. In Settings > Safari, tap Clear History and Website Data, and then tap Clear History and Data. It doesn’t offer a duration selector.

Glenn Fleishman

Erase cookies and website data from within Safari, but you can’t dump all the caches that might harbor evercookies.

(You can delete just Web site specific data: tap Settings > Advanced > Website Data, wait for items to load, then swipe to the bottom. Tap Remove All Website Data. However, this will leave browser cookies in place.)

Because iOS doesn’t have an option to empty all caches, you’re at greater risk for evercookie tracking, which means it’s more sensible to use private browsing with sites and services with which you’re less familiar.

Since Kamkar’s original post in 2010, over-crafty tracking firms have [found other ways](http://www.macworld.com/article/3105005/os-x/privacy-problems-on-the-web-even-your-devices-battery-life-can-be-used-to-track-you.html) that may have a statistical chance of success in identifying you. Browser makers seem more concerned about these and have patched some of the ways in which private or browser-identifiable information leaks.

**Look! Down in the gutter! It’s a turd! It’s explained!**

The evercookie likes to respawn itself, but supercookies want to be unavoidable. At least two major Internet providers, [AT&T](https://www.propublica.org/article/att-stops-using-undeletable-phone-tracking-ids) and [Verizon Wireless](https://www.eff.org/deeplinks/2014/11/verizon-x-uidh), inserted for a time a global identifier into users’ Web requests that allowed them to be tracked by their Internet account across *everything*. It was a terrible idea and badly implemented, because ad networks and malicious parties could uniquely ID customers and customer networks through an unchanging number.

While both ISPs stopped using supercookies, there’s no prohibition against them. The issue was more disclosure and the issue of opting out of such tracking.

But you can prevent any Web-based labeling of your activity through network-based injection by either using a virtual private network (VPN) or ensuring you’re always visiting secure versions of sites. This also avoids networks from inserting pop-up warnings on top of Web pages that aren’t under their control, as they can inject JavaScript or other code, which is a terrible security practice.

For the former, you can use rent-a-VPN services, like [TunnelBear](https://www.tunnelbear.com/) and [Cloak](https://www.getcloak.com/), which offer a variety of time- and data-based subscriptions that let you encrypt everything from your computer or iOS device past local networks out to a VPN server in a data center elsewhere on the Internet. Free VPNs, [like the one from Opera Software](http://www.opera.com/blogs/desktop/2016/04/free-vpn-integrated-opera-for-windows-mac/), are also out there, but check on their privacy policies.

It can be harder to ensure that you’re always on an https connection to a website, but it’s getting easier, and browsers will start signaling more forcefully when you’re not. Ever more websites have realized that even casual data reveals something about their users and that malicious parties may try to inject malware and other nonsense into their connection to a customer. On top of that, a project of the Electronic Frontier Foundation has made it free to obtain the digital certificates required for https.

**More privacy where it counts**

It seems to me that security-minded folks could monitor a JavaScript for all the ways in which it tries to store information, and examine cached data coming down from a Web server. Despite efforts to hide this kind of nonsense, evercookie behavior can be characterized, and it could be blocked or users could be warned.

Browser makers like Apple could have more of an impact on tracking by offering these features. There’s no reason for evercookies to exist: when someone deletes a regular cookie, it’s intentional, and any attempt to circumvent that isn’t in the user’s interest. Right now, you have to employ the many techniques I note above, and we should have an option short of a nuclear one.

*Note: When you purchase something after clicking links in our articles, we may earn a small commission. Read our*[*affiliate link policy*](https://www.macworld.com/about/affiliates.html)*for more details.*