A **zombie cookie** is an [HTTP cookie](https://en.wikipedia.org/wiki/HTTP_cookie) that is recreated after deletion. The term was created by Attorney Joseph H. Malley, who initiated the Super-Cookie Class Actions in 2010. Cookies are recreated from backups stored outside the [web browser](https://en.wikipedia.org/wiki/Web_browser)'s dedicated cookie storage. It may be stored online or directly onto the visitor's computer, in a breach of [browser security](https://en.wikipedia.org/wiki/Browser_security). This recreation mechanism makes zombie cookies very difficult to remove. These cookies may be created on a web browser that has opted not to receive cookies since they do not entirely rely on traditional cookies.

Purpose[[edit](https://en.wikipedia.org/w/index.php?title=Zombie_cookie&action=edit&section=1)]

[Web analytics](https://en.wikipedia.org/wiki/Web_analytics) collecting companies use cookies to track Internet usage and pages visited for [marketing research](https://en.wikipedia.org/wiki/Marketing_research).[[1]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-code-1) Sites that want to collect user statistics will install a cookie from a traffic tracking site that will collect data on the user. As that user surfs around the web the cookie will add more information for each site that uses the traffic tracking cookie and sends it back to the main tracking server.

Zombie cookies allow the [web traffic](https://en.wikipedia.org/wiki/Web_traffic) tracking companies to retrieve information such as previous [unique user](https://en.wikipedia.org/wiki/Unique_user) ID and continue tracking personal browsing habits. When the user ID is stored outside of a single browser's cookie storage, such as in a header injected by the network into HTTP requests, zombie cookies can track users across browsers on the same machine.[[2]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-2)

Zombie cookies are also used to remember unique IDs used for logging into websites. This means that for a user that deletes all their cookies regularly, a site using this would still be able to personalize to that specific user.

## Implications[[edit](https://en.wikipedia.org/w/index.php?title=Zombie_cookie&action=edit&section=2)]

A user that doesn't want to be tracked may choose to decline 3rd party cookies or delete cookies after each browsing session.[[3]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-3) Deleting all cookies will prevent some sites from tracking a user but it may also interfere with sites that users want to remember them. Removing tracking cookies is not the same as declining cookies. If cookies are deleted this causes the data collected by tracking companies to become fragmented. For example, counting the same person as two separate unique users would falsely increase this particular site's unique user statistic. This is why some tracking companies use a type of zombie cookie.

## Implementation[[edit](https://en.wikipedia.org/w/index.php?title=Zombie_cookie&action=edit&section=3)]

According to TRUSTe: "You can get valuable marketing insight by tracking individual users' movements on your site. But you must disclose your use of all [personally identifiable information](https://en.wikipedia.org/wiki/Personally_identifiable_information) in order to comply with the Fair Information Practices guidelines".[[4]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-truste-4)

Possible places in which zombie cookies may be hidden include:

* Standard HTTP cookies
* Storing cookies in and reading out web history
* Storing cookies in [HTTP ETags](https://en.wikipedia.org/wiki/HTTP_ETag)
* Internet Explorer userData storage (starting [IE9](https://en.wikipedia.org/wiki/IE9), userData is no longer supported)
* [HTML5](https://en.wikipedia.org/wiki/HTML5) Session Storage
* HTML5 Local Storage
* HTML5 Global Storage
* HTML5 Database Storage via SQLite
* Storing cookies in RGB values of auto-generated, force-cached PNGs using HTML5 Canvas tag to read pixels (cookies) back out
* [Local shared objects](https://en.wikipedia.org/wiki/Local_shared_object) (Flash cookies)
* Silverlight Isolated Storage
* Cookie syncing scripts that function as a cache cookie and respawn the MUID cookie[[5]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-mayer-5)
* [TCP Fast Open](https://en.wikipedia.org/wiki/TCP_Fast_Open)
* [TLS's Session ID](https://en.wikipedia.org/wiki/Transport_Layer_Security#Session_IDs)

If a user is not able to remove the cookie from every one of these data stores then the cookie will be recreated to all of these stores on the next visit to the site that uses that particular cookie. Every company has their own implementation of zombie cookies and those are kept proprietary. An open-source implementation of zombie cookies, called [Evercookie](https://en.wikipedia.org/wiki/Evercookie),[[6]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-samy-6) is available.

## Controversies[[edit](https://en.wikipedia.org/w/index.php?title=Zombie_cookie&action=edit&section=4)]

In 2015, TURN, an online advertising clearinghouse,[[7]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-propublica.org-7) introduced zombie cookies based on Flash Local Shared objects.[[8]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-informationweek-8) Privacy advocates quickly denounced the technology.[[9]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-epic-9)

An academic study of zombie cookies was completed in 2009, by a team of researchers at [UC Berkeley](https://en.wikipedia.org/wiki/UC_Berkeley),[[10]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-:0-10) where they noticed that cookies which had been deleted, kept coming back, over and over again. They cited this as a serious privacy breach. Since most users are barely aware of the storage methods used, it's unlikely that users will ever delete them all. From the Berkeley report: "few websites disclose their use of Flash in privacy policies, and many companies using Flash are privacy certified by TRUSTe".[[10]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-:0-10)

Ringleader Digital made an effort to keep a persistent user ID even when the user deleted cookies and their HTML5 databases. The only way to opt-out of the tracking, was to use the company's opt-out link, which gives no confirmation.[[11]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-arstechnica-11) This resulted in a lawsuit against Ringleader Digital.

The term *"zombie cookie"* was created by Attorney Joseph H. Malley who initiated the Super-Cookie Class Actions in 2010.[[*promotional language*](https://en.wikipedia.org/wiki/Wikipedia:NOT#SOAPBOX)] The etiology of the phrase was derived from his prior research into Apple's third-party iPhone applications. Some of these which had been criticized as being *"zombie-like"* applications such as the *"super-cookies"* which *"re-spawned"* when deleted. Attorney Malley envisioned a cookie that seemed to come back from the *"dead"*. Blending the two ideas, he first coined the phrase Zombie Cookies within his filed Class Actions, as a means to enable the court, jury, and public understand the basis of the litigation.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

The Zombie Cookie lawsuits were filed suit in the [United States District Court for the Central District of California](https://en.wikipedia.org/wiki/United_States_District_Court_for_the_Central_District_of_California) against [Quantcast](https://en.wikipedia.org/wiki/Quantcast), [Clearspring](https://en.wikipedia.org/wiki/Clearspring), [VideoEgg](https://en.wikipedia.org/wiki/SAY_Media), and affiliated sites owned by [Walt Disney Internet Group](https://en.wikipedia.org/wiki/Walt_Disney_Internet_Group), [Warner Bros.](https://en.wikipedia.org/wiki/Warner_Bros.) and others. According to the charges, Adobe [Flash cookies](https://en.wikipedia.org/wiki/Flash_cookies) are planted to "track Plaintiffs and Class Members that visited non-Clearspring Flash Cookie Affiliates websites by having their online transmissions intercepted, without notice or consent".[[12]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-out-law-12)

Two "[supercookie](https://en.wikipedia.org/wiki/HTTP_cookie#Supercookie)" mechanisms were found on Microsoft websites in 2011, including cookie syncing that respawned MUID cookies.[[5]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-mayer-5) Due to media attention, Microsoft later disabled this code.[[13]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-13)

Consumer outrage related to Flash cookies and violation of consumers' privacy caused U.S. Congressional Hearings, led by Senators [Al Franken](https://en.wikipedia.org/wiki/Al_Franken) and [John Rockefeller](https://en.wikipedia.org/wiki/Jay_Rockefeller). Reportedly, the "Zombie Cookie", aka Flash Cookie filings, forced [Adobe Systems Inc.](https://en.wikipedia.org/wiki/Adobe_Inc.) to stop processing flash cookies on 98% of all consumers' computing devices.[[14]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-14)

The online advertising clearinghouse TURN implemented zombie cookies on [Verizon](https://en.wikipedia.org/wiki/Verizon) mobile phones, using a hidden, unremovable number by which Verizon could track customers. After an article by [ProPublica](https://en.wikipedia.org/wiki/ProPublica) revealed this fact in January 2015, TURN claimed it had suspended usage of their zombie cookies.[[7]](https://en.wikipedia.org/wiki/Zombie_cookie#cite_note-propublica.org-7)

You can test whether the header is injected in your traffic by visiting [lessonslearned.org/sniff](http://lessonslearned.org/sniff) or [amibeingtracked.com](http://www.amibeingtracked.com/) over a cell data connection.

### How X-UIDH Works, and Why It's a Problem

Like a cookie, this header uniquely identifies users to the websites they visit. Verizon adds the header at the network level, between the user's device and the servers with which the user interacts. Unlike a cookie, the header is tied to a data plan, so anyone who browses the web through a hotspot, or shares a computer that uses cellular data, gets the same X-UIDH header as everyone else using that hotspot or computer. That means advertisers may build a profile that reveals private browsing activity to coworkers, friends, or family through targeted advertising.

Also unlike a cookie, Verizon's header is nearly invisible to the user and can't be seen or changed in the device's browser settings. If a user clears their cookies, the X-UIDH header remains unchanged. Worse, ad networks can immediately assign new cookies and link them to the cleared cookies using the unchanged X-UIDH value. We don't know which data brokers and ad networks are using the header to create behavioral profiles, but [Cory Dunne found](http://codydunne.blogspot.com/2014/10/verizon-wireless-injecting-tracking.html) at least one GitHub repository contained code to extract the header value, as of October 27. The repository has since been quietly deleted but [can be viewed at the Internet Archive](https://web.archive.org/web/20141027194059/https:/github.com/Funnerator/fast_tim_conf/blob/master/lua/id_set.lua). Twitter's mobile advertising division also [appears to use the header for ad auctions](http://www.propublica.org/article/somebodys-already-using-verizons-id-to-track-users).

Besides cookie clearing, the X-UIDH header bypasses several other built-in browser privacy mechanisms. Cookies belong to a single website and aren't shared with other websites. But one unique X-UIDH header value is shared with all unencrypted websites a user visits, making it easier for ad networks to track that user across many sites in a way not possible with cookies alone. Browsers provide Incognito Mode or Private Browsing Mode in order to defeat some kinds of tracking, but the X-UIDH header, since it is injected at the network layer, ignores those modes. Verizon also chooses to ignore [Do Not Track](https://www.eff.org/issues/do-not-track), a setting users enable in their browser to indicate they do not want to be tracked. Similarly, disabling third-party cookies in browser settings does nothing to stop the X-UIDH header.

To compound the problem, the header also affects more than just web browsers. Mobile apps that send HTTP requests will also have the header inserted. This means that users' behavior in apps can be correlated with their behavior on the web, which would be difficult or impossible without the header. Verizon describes this as a key benefit of using their system. But Verizon bypasses the 'Limit Ad Tracking' settings in iOS and Android that are specifically intended to limit abuse of unique identifiers by mobile apps.

Because the header is injected at the network level, Verizon can add it to anyone using their towers, even those who aren't Verizon customers. Notably, Verizon appears to inject the X-UIDH header [even for customers of Straight Talk](https://twitter.com/RonnicaZ/status/525786331272998912), a mobile network reseller (known as a MVNO) that uses Verizon's network. Customers of Straight Talk don't necessarily have a relationship with Verizon.

But according to AdAge, "[Corporate and government subscribers are excluded from the new marketing solution](http://adage.com/article/digital/verizon-target-mobile-subscribers-ads/293356/)." We haven't verified (and Verizon refuses to say) whether the header is still sent for those subscribers or not. If they are indeed excepted from the program, that indicates to us that implementing an opt-out is feasible. We're disappointed that Verizon takes some of its users' privacy more seriously than others.

### Verizon's Claimed Protections

Verizon does provide a sort of limited [opt-out](https://www.verizonwireless.com/myprivacy/) for individual customers, but it appears that the opt-out does not actually disable the header. Instead, it merely tells Verizon not to share detailed demographic information with advertisers who present a UIDH value. Meaningful protection from tracking by third parties would require Verizon to omit the header entirely.

According to Verizon, the header value is a [salted hash](https://security.stackexchange.com/questions/51959/why-are-salted-hashes-more-secure), and the hash changes [on an undisclosed frequency](http://www.propublica.org/article/somebodys-already-using-verizons-id-to-track-users). However, it's easy for third-party ad networks to create a continuous profile by associating old and new X-UIDH values through their own identifier cookie[1](https://www.eff.org/deeplinks/2014/11/verizon-x-uidh#footnote1_4eoq25w). Verizon has refused to say what identifier they hash to create the identifier, but [their recent patent](http://patents.justia.com/patent/8763101) suggests hashing a phone number. If they are indeed hashing phone numbers, it would be a major cryptographic mistake. Phone numbers can easily be deduced from hashes, so sending those hashes to untrusted web sites is practically equivalent to giving them your phone number.

Besides the ad networks, the unique X-UIDH header is a boon to eavesdroppers. We have seen that the NSA uses similar identifying metadata as 'selectors' to collect all of a single person's Internet activity. They also have been shown to use selectors to choose targets for delivering malware via [QUANTUMINSERT](http://www.wired.com/2013/11/this-is-how-the-internet-backbone-has-been-turned-into-a-weapon/) and similar programs. Having all Verizon mobile users' web traffic marked with a persistent, unique identifier makes it trivial for anyone passively eavesdropping on the Internet to associate that traffic with the individual user in a way not possible with IP addresses alone.

According to Verizon, it [began the Precision Market Insights program in 2012](http://www.huffingtonpost.com/2012/10/17/verizon-precision-market-insights_n_1971265.html), but has consistently refused to provide technical details about how the program worked. The injection of the X-UIDH header went largely unremarked by the technical community until recently because it is so hard to observe. The header is inserted in requests after they leave the phone, so customers cannot detect it using only a phone. In order to detect it, a user needs to run a web server configured to log or echo all HTTP headers, which is very rare.

### How You Can Protect Yourself

Verizon can only modify plaintext traffic. It can't modify encrypted requests without breaking the whole connection. There are four options for encrypting web requests: HTTPS, an encrypted proxy, a VPN, or Tor. Only a VPN or Tor provide full protection in this case.

The best protection against this specific problem is to use a [VPN](https://ssd.eff.org/en/module/choosing-vpn-thats-right-you) that encrypts all requests made from your phone, regardless of whether they were made by an app or a browser. Most VPNs are paid services, and when using a VPN you have to trust the VPN operators the same way you would normally trust your ISP. Advanced users can also use [Tor](https://www.torproject.org/) via [Orbot Android app](https://play.google.com/store/apps/details?id=org.torproject.android&hl=en) in transparent proxy mode (requires root). Tor is free, but you have to trust exit node operators [not to interfere with your connection](http://www.leviathansecurity.com/blog/the-case-of-the-modified-binaries/). Tor is more appropriate if you are trying to be anonymous.

The second-best protection is to use an encrypted proxy, which protects browser traffic but not mobile apps. Mobile Chrome provides the '[Reduce data usage](https://support.google.com/chrome/answer/2392284?hl=en)' setting, which is reported to prevent the X-UIDH header injection. Unfortunately, this connection is not reliably encrypted, because [an ISP can disable encryption on it at any time](https://support.google.com/chrome/answer/3517349?hl=en).

HTTPS, which is the best protection for many types of harm, is actually the least powerful protection for this one. The header cannot be injected into an HTTPS request, but since websites choose whether to offer HTTPS, a site that wants to track users can simply avoid HTTPS and get the tracking headers. [The web needs to become fully encrypted](https://www.eff.org/deeplinks/2013/11/encrypt-web-report-whos-doing-what), and these X-UIDH headers provide a strong disincentive for sites and advertisers who wish to track their users to adopt HTTPS. In fact, the [AT&T patent](https://www.google.com/patents/US20130273886) on similar headers recommends downgrading (redirecting) secure HTTPS requests to HTTP ones in order to receive the tracking header.

### What Verizon Should Do

Verizon should immediately stop injecting the X-UIDH tracking header into its users' traffic. It is entirely possible to re-design their marketing programs so that the header is only injected for users who explicitly consent to having their Internet connections modified to add tracking information, and to do so in a way that [doesn't allow third-party sites to track users across the Internet](http://webpolicy.org/2014/10/24/how-verizons-advertising-header-works/).

We're also concerned that Verizon's failure to permit its users to opt out of X-UIDH may be a violation of [the federal law](http://www.law.cornell.edu/uscode/text/47/222) that requires phone companies to maintain the confidentiality of their customers' data. Only two months ago, the wireline sector of Verizon's business was hit with a [$7.4 million fine](http://transition.fcc.gov/Daily_Releases/Daily_Business/2014/db0903/DOC-329127A1.pdf) by the Federal Communications Commission after it was caught using its "customers' personal information for thousands of marketing campaigns without even giving them the choice to opt out." With this header, it looks like Verizon lets its customers opt out of the marketing side of the program, but not from the disclosure of their browsing habits.

More generally, Verizon should stop tampering with their customers' Internet traffic without their customers' consent. ISPs like Verizon act as trusted connectors to the world, and shouldn't be modifying our communications on their way to the Internet. People should not be required to subscribe to a VPN and put their trust in a third party in order to get a modicum of privacy on the Internet.

AT&T has been [reported to be testing a similar header](http://www.forbes.com/sites/kashmirhill/2014/10/28/att-says-its-testing-unkillable-tracker-on-customers-smartphones/).

[1.](https://www.eff.org/deeplinks/2014/11/verizon-x-uidh#footnoteref1_4eoq25w)For instance, suppose an ad network assigned you a cookie with the unique value "cookie1," and Verizon assigned you the X-UIDH header "old\_uid." When Verizon changes your X-UIDH header to a new value, say "new\_uid," the ad network can connect "new\_uid" and "old\_uid" to the same cookie value "cookie1" and see that they all three values represent the same person. Similarly, if you subsequently clear cookies, the ad network will assign a new cookie value "cookie2." Since your X-UIDH value is the same (say, "new\_uid") before and after clearing cookies, the ad network can connect "cookie1" and "cookie2" to the same X-UIDH value "new\_uid." The back-and-forth bootstrapping of identity makes it impossible to truly clear your tracking history while the X-UIDH header is enabled.