Browser Vendors: fight for your users

Thomas Lowenthal
Center for Information Technology Policy
Princeton University
tlowenth@princeton.edu

Abstract

The parties who track users online are technically sophisticated, dedicated, and motivated by significant financial gains. Users often lack the technical knowledge to understand the forms of tracking that are deployed against them, the skills necessary to deploy countermeasures, or the significant quantities of time and effort necessary to safeguard their privacy. Browser vendors, on the other hand, have the resources, capacity, and expertise necessary to protect their users from many different privacy threats. Browser vendors should take responsibility for their role as users' agents online and use their technical and market power to protect user interests.

1 The Browser: Users' Privacy Trust Root

The web browser is literally the user's representative online. As the user's agent, a browser should act based on the needs of the user; design policy decisions should be based exclusively on the user's priorities. Indeed, the browser is the only party that the user should have to rely upon to work for them: it's much easier to make a one-time trust judgment about which web browser to use than it is to have to make repeated, ongoing, granular trust judgments about numerous websites, and their embedded and active content.

Indeed, it would be prohibitive to expect users to audit the potential privacy risks posed by the embedded web bugs, persistent & novel cookies, JavaScript content, tracking practices, and information sharing policies of all the many sites they visit. The much more reasonable model has the user choose a trustworthy browser, learn about its security and privacy features, customize individual settings, and then confidently rely that the browser will work to protect them in the choices that they've made, and will make ongoing operational decisions based on the user's expressed preferences.

Browser vendors should protect their users by making privacy-by-design a priority the same way that they do with security. In addition, browsers should be honest with their users, explaining their strengths and weakness, so that users can make informed activities about their activities online. What follows is a selection of ways that browsers currently fail to protect their users' privacy. The difficulty of mitigating or fixing these problems varies, but browser vendors should consider these issues — and others like them — to be important ways that they can protect (or fail to protect) their users. Because of the browser's unique position in users' web-browsing trust hierarchies, these issues demand fixes at the browser level.

2 Web Privacy Weaknesses & Countermeasures

2.1 Cookie & Active Tracking Control

Most users are aware of HTTP cookies, and some are aware of other active tracking measures like flash cookies. However, there are many[7] active tracking measures that can be used to identify and re-identify users. Many of these were not even designed as identification technologies, but result from the 'generous' set of features available among the variety of browser and active content technologies available on the web.

Given how much of our lives we spend online, persistent and pervasive tracking poses a direct threat to individual privacy. It's not that tracking technologies are inherently wrong, far from it. Rather browsers should offer users the technical capacity to choose which sites know and retain what information about them, over which sessions. Defaulting to letting sites keep persistent, hard-to-remove track of users is a mistake: tracking should be an option that's up to the user, and under their control

2.2 Fingerprint Uniqueness Reduction

Even when not using active tracking methods like cookies, passive tracking methods often allow for accurate re-identification of a particular browser. According to the data produced by the EFF's Panopticlick project[3], browsers' fingerprints have an average anonymity set size larger than 280,000, and browsers supporting Flash or Java are 94.2% likely to be unique. However, in the custom browser deployed by the Tor Project, the measures taken to create a uniform browser fingerprint were quite successful, producing highly uniform anonymity sets.

There are lots of trivial steps that browser vendors can take to protect against this method for identifying users. Reporting a slightly more granular browser version number like "1.6" rather than "1.6.0.17" immediately makes fingerprints more homogeneous. Likewise, sorting supported font lists before reporting them takes away another significant source of entropy. These are just some changes made based on the entropy data. Browser vendors have the ability to reconsider the amount of information they really *need* to report to sites. Defaulting to reporting everything may be somewhat sensible in a fragmented, browser-dependent web. However, in a web built on agreed standards, privacy should be the default, with exceptions made for specific information when needed.

2.3 Effective Private Browsing Modes

Most of the modern browsers feature private browsing modes, but research from Stanford University[1] suggests that they may not be well-implemented to provide the sort of privacy protections that users might expect. In addition to exploitable weaknesses which may allow traces to be left locally after private browsing, these modes fail to implement the anonymity measures which would be required to prevent a hostile website from associating non-private browsing with a series of distinct private browsing sessions.

Private browsing modes are an important tool in a users's privacy defense arsenal. They allow users to retain control of their personal information in ways which might not otherwise be possible. They may even permit users to engage in behavior which they might otherwise have considered too risky. As such, it's imperative that these modes are effective, and live up to users' functionality expectations.

2.4 History Retrieval

It has for some time been possible to use cunningly crafted HTML & CSS to infer users' complete browsing history[6], which may contain all kinds of sensitive information, and — moreover — makes for a fairly unique way to re-identify the same users. This is mentioned less to draw attention to this particular attack, and more as a comment on these sorts of browser weaknesses. As long as browser vendors leave this sort of gaping vulnerability unchecked, their users will continue to be at risk.

The problem is that the drive to patch privacy holes doesn't seem to be nearly as strong as the drive to fix security holes, or developing new and innovative features. However, for many users, improved privacy protection is much more valuable than shiny new tab-sorting features. While competitions like Pwn2Own glamourize and reward security development, privacy design often plays second fiddle.

2.5 Certificate Trust Control

As recent events[2][8][5] and commentary[10][4][9] have indicated, the public-key identification infrastructure which underpins our web encryption technology is hopelessly broken. This failure isn't a technical one, it's a social one, and browser vendors are at least partly to blame. There have been no movements to revoke the signing powers of the several certificate authorities which fail. Users rely on the the security practices of every single certificate authority whenever they do online banking, or transfer personal medical information online. When a CA spectacularly fails, a browser vendor should pro-actively call them on it, acting on the trust that users place in their browser by revoking the CA's authority.

Yes, these sorts of aggressive enforcement actions 'break' some sites. However, that should be the desired behavior. When the browser represents to the user that a secure connection is taking place, it should be on the basis of that actually being true. If a CA is failing their authentication responsibility, the browser should not mislead the user by asserting that everything is hunky-dory when an attack may actually be taking place.

3 Conclusion

The browser is the user's only intermediary and protector from the dangerous ravages of a cold, dark, unfriendly web. It is practically the case that web services lust after users' personal information, extended click- and browsing-history, and mostly succeed in getting it. A browser sits as the root of a user's trust tree, and has a unique responsibility to safeguard the user's privacy interests online.

External policy measures like Do Not Track, data breach notifications, privacy policies, and personal information protection laws are valuable, but they have their limitations. Laws are hard to enforce across borders; privacy policies are incredibly difficult to read and even harder for users to verify or audit. The best way to keep information from being used against the user is to prevent it from leaking out in the first place. That begins and ends with the browser.

References

[1] Gaurav Aggarwal et al. "An Analysis of Private Browsing Modes in Modern Browsers". In: *USENIX 2010*. 2010. URL: `http://crypto.stanford.edu/~dabo/pubs/papers/privatebrowsing.pdf`.

[2] Jacob Appelbaum. "Detecting Certificate Authority compromises and web browser collusion". In: *The Tor Blog* (Mar. 22, 2011).

[3] Peter Eckersley. "How Unique Is Your Web Browser?" In: *Privacy Enhancing Technologies Symposium (PETS 2010)*. Ed. by Electronic Frontier Foundation. 2010. URL: `https://panopticlick.eff.org/browser-uniqueness.pdf`.

[4] Ed Felten. "Web Certification Fail: Bad Assumptions Lead to Bad Technology". In: *Freedom to Tinker* (Feb. 23, 2010). URL: `http://www.freedom-to-tinker.com/blog/felten/web-certification-fail-bad-assumptions-lead-bad-technology`.

[5] Phillip Hallam-Baker. "The Recent RA Compromise". In: (Mar. 23, 2011). URL: `http://blogs.comodo.com/it-security/data-security/the-recent-ra-compromise/`.

[6] Artur Janc and Lukasz Olejnik. "Feasibility and Real-World Implications of Web Browser History Detection". In: *Web 2.0 Security and Privacy 2010*. 2010. URL: `http://w2spconf.com/2010/papers/p26.pdf`.

[7] Samy Kamkar. *Evercookie*. URL: `http://samy.pl/evercookie/`.

[8] Steve Schultze. "Web Browsers and Comodo Disclose A Successful Certificate Authority Attack, Perhaps From Iran". In: *Freedom to Tinker* (Mar. 23, 2011). URL: `http://www.freedom-to-tinker.com/blog/sjs/web-browsers-and-comodo-disclose-successful-certificate-authority-attack-perhaps-iran`.

[9] Steve Schultze. "Web Security Trust Models". In: *Freedom to Tinker* (Feb. 22, 2010). URL: `http://www.freedom-to-tinker.com/blog/sjs/web-security-trust-models`.

[10] Christopher Soghoian and Sid Stamm. "Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL". In: *SSRN* (Apr. 14, 2010). URL: `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1591033`.