Identity in the Browser: Easy Wins and Guiding Principles

Naveen Agarwal, Miranda Callahan, Tyler Close, Travis McCoy, Chris Messina, Glen Murphy, and Dirk Pranke

The W3C has called for a workshop on the role the web browser can play in online identity management, and suggests a focus on solutions that are “effective enhancements to Web browsers that lead to trustworthy benefits that can be realized in the near term” [1]. The authors believe that this is an important topic and one that is consistent with the goals they have been pursuing in Chromium in recent months. Please note that what follows is solely the opinion of the authors and should not be construed as an official statement of the Chromium project, Google Chrome, or Google as a whole.

Part I: How can the browser help?

As the W3C’s call outlined, most recent activity in online management has been centered around solutions that assume the browser cannot add new functionality. Over the past decade, the slow pace of browser evolution made this a reasonable assumption, but recently the pace of innovation has quickened dramatically across all of the major browser vendors and it is fair to revisit this design principle.

In particular, we note that the browser has two important functions that make it relevant.

First, it is perhaps a tautology that the browser is involved in every interaction on the web. No matter how much great work any given server-side-only solution might do, it will only cover the websites that the solution can directly (or indirectly) influence, and large chunks of the web will remain unaffected. The browser, on the other hand, can provide total coverage.

Second, the browser is by definition a trusted component. If you want to view website X in your browser, you have to trust the browser to display it correctly, and to not take unintended actions with your data while doing so.

We also note that while it is possible to extend most browsers today through add-ons or extensions, and that there are many good extensions to managing passwords and other identity-related improvements, experience shows that the vast majority of users do not use these extensions (presumably because they are unaware of them), and so building the functionality directly into the browser can make a big difference.

Of course, it is also important to realize that there are important drawbacks to a browser-based solution.

For example, again almost a tautology, any solution will only work in a particular web browser instance. If you use the same browser on multiple machines, unless something is done they cannot share data. In some cases, this could be a good thing (isolating sensitive corporate data from a personal machine), but in most cases, this will simply be annoying.

Next, it seems that fewer and fewer of us use only one kind of brand of browser. At the very least, many of us now browse the web from tablets and phones as well as desktops, laptops. And even on a single computer, it appears that users are increasingly becoming aware of the variety of browsers and switching between them; the days when you used the single browser

that came installed on your machine appear to be ending.

And, of course, browsers must support existing web sites, and web sites must support existing browsers.

These points suggest that while a *browser-only* solution is probably not a great idea, a solution that takes advantage of what can be done in a browser could be better than one that doesn't.

Part II: Chromium's biases

The initial developers of Chromium explicitly focused on "speed, simplicity, security, and stability"[1]. Let's look at how these principles might apply to online identity.

Speed suggests that good features for managing your identity should *speed up* your browsing experience. Using a password manager can save you keystrokes and clicks. Supporting multiple identities (or profiles) in a single browser instance can keep you from having to log out and log back in, or exit and restart the browser, or even from logging out of your computer in order to log back in in a different account. All of these can improve the user's experience without requiring any changes to the ecosystem of the web.

The security implications are obvious. We know that passwords can be weak links on the Internet. We know that anything that can be done to limit the number of times a user may accidentally enter in credentials for site A onto site B would be good, and that the best way to do this is to not require the user to enter anything at all. The more the browser can become aware of what credentials the user might want to use on a given site, and help with that interaction, the better.

In addition, it is important to realize that identity is a double-edged sword for security. In some cases, strong authentication can make you more secure, but in some situations you wish to be as anonymous as possible. Chromium's early support for Incognito mode, which makes it easy for the user to temporarily isolate one set of activities from another, has helped preserve anonymity as a core feature of a web browser.

However, perhaps the most interesting aspect to consider is simplicity, for it is clear that complexity is a huge barrier to effective online identity management. Managing lots of passwords is complex. Power users have an even more complex lot, because they frequently have to manage multiple identities for a single service. It is also fairly easy to design complex federated systems that either expose highly technical details to the user (initial deployments of OpenID and WebFinger that expose URLs as authentication endpoints) or build systems that have such complex mental models that it's hard to tell what they will even do (examples include SAML and permissions-granting schemes for application data sharing).

Here Chromium has taken a fairly conservative approach, to limit what we will do to what can be done simply and easily. For example, our current work on supporting multiple profiles is fairly limited - a user will be able to have two windows open, each managing a separate set of credentials (e.g., a separate set of saved usernames and passwords for multiple sites, separate

---

[1] There doesn't appear to be a publicly citable source for this, but the http://www.google.com/chrome/intl/en/make/features.html?brand=CHKZ cites "speed, simplicity, and security". Hopefully we still care about stability, too.

cookie jars, etc.). This will give us strong guarantees of stability and security -- credentials cannot accidentally get leaked from one page to another -- and has a clear user model, but this comes at the cost of some more advanced configurability and features. You cannot easily have multiple identities in a single browser window, for example, and you cannot build complex mappings and rules for when to use a given credential in a given profile; instead, profiles work just as regular browsers do.

In addition, providing synchronization of credentials across browser instances can make things simpler for the user.

Opportunities for the future

It is clear that we are not where we'd like to be. It is too easy to be locked into a given product, or a given service provider. We strongly believe that identity management should be standardized to avoid vendor lock in. It should be easy for me to migrate my credentials from browser A to browser B, if that is what I want to do. It should be easy for me to reuse the identities and accounts I want to reuse across the web, rather than being required to use either a profile owned by the service provider or a select number of partners.

It is also clear that there are different levels of anonymity and privacy that should be easily available. As Google outlined in a recent blog post [2], the web should continue to support anonymous and pseudonymous experiences as well as fully authenticated experiences. When the user chooses to not be fully identified, it is the browser's job to make sure it prevents identity from leaking out accidentally (at least to the extent it can do so).

Finally, it is clear that existing solutions are unmanageable, and unmanageably different from each other. Why are there umpteen different sets of password requirements for different sites? How is a user supposed to manage them all and hope to still choose secure passwords? How is the user supposed to keep track of which sites have been granted access to their data, let alone control subsets of that data, or revoke it?

Yet while we acknowledge these difficulties, we must strive to build tools that make our lives simpler and faster as well as more secure. One out of three isn't good enough.

References

[1] http://www.w3.org/2011/identity-ws/
[2] http://googlepublicpolicy.blogspot.com/2011/02/freedom-to-be-who-you-want-to-be.html