Goodbye Short Sessions: A Proposal for Using Service Workers to Improve Cookie Management on the Web

By William Denniss

Google Identity Product Manager

By Owen Campbell-Moore

Owen is a contributor to Web**Fundamentals**

We all love how native apps will ask you to login only once and then remember you until you tell them you want to log out. Unfortunately the web doesn't always work that way.

Now that devices, especially mobile devices, are more personal, and with more sites sending all traffic over HTTPS reducing the risk of token theft, websites should reconsider their short-lived cookie policies and adopt more user-friendly longer lived sessions.

However, even if you *want* to make the session last longer, some websites don't verify the user's authentication on each request (in other words, there is no way to revoke the session cookie once issued). This normally leads to short sessions, with the user being forced to sign in frequently so their authentication can be re- validated, allowing for things like a password change to invalidate existing sessions in a known amount of time.

If this is an approach that you use, we have a technical solution that may help you automatically re-validate the stateless authentication cookie. It works by having a secondary long-lived token that can be used to refresh your existing short-lived authentication cookie. Leveraging the new service worker pattern allows us to regularly "check in" with the long lived token, verify the user's authentication (for example, check to see if they have not recently changed their passwords, or otherwise revoked the session) and re-issue a new short-lived authentication cookie.

A practical proposal for migrating to safe long sessions on the web

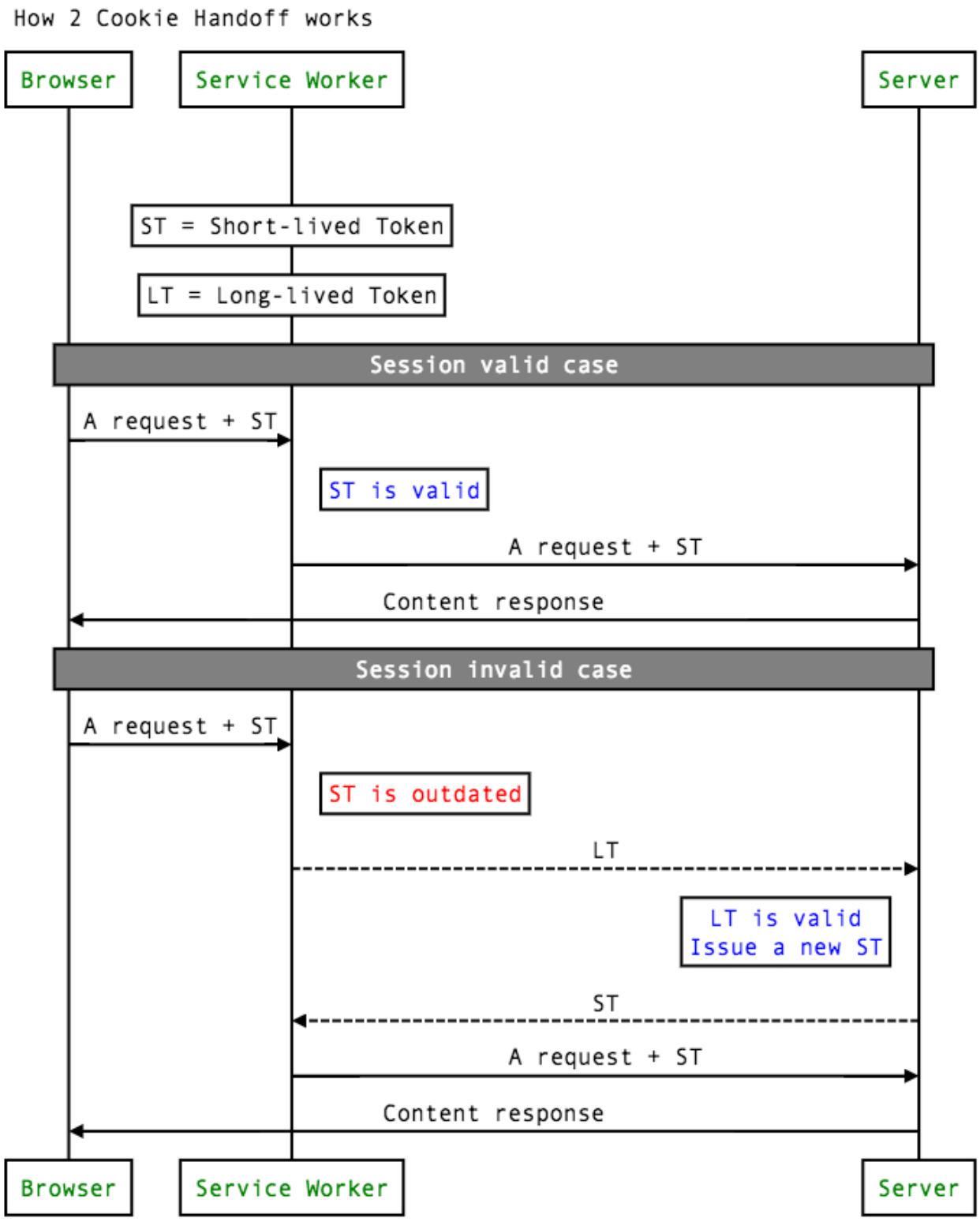
From here, this post describes a new technique we're proposing that we call **2-Cookie-Handoff (2CH)**. We are hoping to use this article to hear community feedback on whether

this approach seems positive, and if so to work with the industry on documenting best practices for using 2CH.

Service workers are a new technology supported by multiple browsers such as Chrome, Firefox, Opera and coming soon to Edge. They allow you to intercept all network requests from your site through a common point of code on the client, without modifying the existing pages. This allows you to set up a "2CH worker" for logged in users that can intercept all of the network requests your page is making and perform the token swapping just like mobile apps do.

Much of the time your server already has an endpoint used by mobile apps to obtain a new short-lived token, typically using the OAuth protocol. To enable the above pattern on the web, that endpoint just needs to be updated to understand when it is being called by a service worker, and then return a new short-lived session cookie formatted in a way that other pages on the site already expect.

If your server doesn't already have such an endpoint, it can create one just for browser session management.



The two-token pattern with service workers follows the OAuth 2.0 pattern fairly closely, if you already run an OAuth token endpoint, then you can likely re-use it with service workers for your web authentication.

You may also be wondering what happens if the user visits a browser that doesn't support service workers. If you implement the above approach they will simply experience no

difference and continue to experience short sessions.

We have published a sample client and backend. We hope you will try it for yourself and answer a survey about session management.

Last updated July 2, 2018.