work, so they don't need to sign back in every time they click a new link. However, in many cases, websites and third parties have used cookies to surreptitiously track users and their browsing habits. The user community responded to this privacy threat in multiple ways. A 2011 cookie-retention study by comScore showed that approximately one in three users deletes both first- and third-party cookies within a month after visiting a website.[1] Also, multiple browser extensions are available that can reveal third-party tracking, such as Ghostery (www.ghostery.com), as well as "hidden" third-party affiliations between sites, such as Lightbeam (www.mozilla.org/en-US/lightbeam). In addition, modern browsers now have native support for the rejection of all third-party cookies, and some even enable it by default.

to their browsing histories. Jonathan Mayer[2] and Peter Eckersley[3] showed that the features of a browser and its plug-ins can be fingerprinted and used to track users without cookies. Today, a few commercial companies use such methods to provide device identification through Web-based fingerprinting. According to Keaton Mowery and his colleagues' classification, fingerprinting can be used either constructively or destructively.[4] Constructively, a correctly identified device can combat fraud, for example, by detecting that a user trying to log in is likely an attacker using stolen credentials or cookies. Destructively, device identification through fingerprinting can be used to track users between sites without their knowledge and without a simple way to opt out.

Copublished by the IEEE Computer and Reliability Societies

Companies offered fingerprinting services as early as 2009, and experts were already voicing concerns over their impact on user privacy.[5] Even when fingerprinting companies honor the recently proposed Do Not Track (DNT) header, users are still fingerprinted for fraud detection, but companies promise not to use the information for advertising purposes.[6] Note that because the fingerprinting scripts execute regardless of the DNT value, verifying this promise is much harder than verifying DNT's effects on stateful tracking, where the effects are visible at the client side, in a user's cookies.[7]

In this article, we survey the current state of practice of Web-based device fingerprinting. First, we analyze the fingerprinting code of three large, commercial companies, then we measure the adoption of fingerprinting

list of fonts from the user's machine. According to his experiment, fonts are one of the most identifying attributes of a user's browsing environment, second only to the list of plug-ins. Fonts tend to be machine specific because natively installed applications tend to use custom fonts that are usually made available systemwide, when the applications are installed.

Eckersley had to rely on third-party plug-in functionality because there's no standard way to obtain font information through JavaScript. Flash and Java, two popular third-party plug-ins, expose API methods for obtaining a list of fonts, presumably to allow for proper rendering of content for applications relying on less-standard font families. However, note that modern