The Mobile Browser as a Web-Based Platform for Identity

Ben Adida*

Abstract

Mobile devices have long been considered useful in bootstrapping authentication via other channels, including the web. As mobile devices begin to include complete web browsers, there is an opportunity to standardize a simple, web-based mobile authentication technique, both for devices themselves and for desktop access using the mobile device as a secondary channel. The path to better authentication will require a standard JavaScript browser API that web sites can easily trigger to achieve simple, secure authentication.

The Need: Easy, Secure Authentication. Authentication on the web is overwhelmingly achieved with passwords, even though it is common knowledge that passwords are woefully insufficient because of phishing [3]. Even discounting the security threat, keeping track of dozens of passwords is particularly inconvenient: users are so averse to new site registration that startup investors and web designers often recommend some level of "account-less" operation to lower the barrier to first use.

Two major solutions have been proposed. Microsoft is implementing CardSpace [2], a browser and operating system extension that presents a particular user login experience using a wallet-and-identity-card metaphor. A separate team is building OpenID [5], a system that uses web-based redirection to achieve single sign-on without any client-side extension.

The Problems: Phishing & Client-Side Deployment. Two major problems arise from proposed web-based authentication solutions, one when no client-side extension is used, one when a client-side extension is introduced.

Without a client-side extension, authentication, even when centralized on a single OpenID site, remains based on a username and password. Users certainly gain in convenience: one password unlocks all sites. On the flip side, this single password becomes extremely valuable and potentially a strong target of phishing [1, 4]. Because current browsers do not provide a special login ritual interface, it is unclear how any vanilla-browser solution might move beyond passwords and ever achieve phishing-resistant authentication.

With a client-side extension, users must upgrade their browsers and, more importantly, ensure that every browser they ever use is updated with this extension. Therein lies the conundrum of Microsoft CardSpace: how will users authenticate from Internet cafes or using a friend's computer?

[*] Children's Hospital Informatics Program, Harvard Medical School, and the Center for Research on Computation and Society at Harvard University. Email: `ben.adida@childrens.harvard.edu / ben@eecs.harvard.edu`.

The Opportunity: Mobile Browsers and Identity Integration. Mobile devices are typically used for telephone conversations tied to a user account. In other words, simply by virtue of operating a mobile device, the user is authenticated to the carrier. Thus, many have proposed using the mobile device as a secure authentication channel. Early proposals called for one-time passwords to be sent via SMS the mobile phone [6], or for simple acknowledgment via SMS [8], effectively using reception ability as authentication. Another approach, in the medical field, proposed using public-key cryptography with the secret key stored on the cell phone's SIM card [7]. There are many similar, custom-protocol alternatives to this same problem.

We propose to extend this thinking to the latest design of mobile devices, such as the Apple iPhone or the Nokia 800: once mobile devices are equipped with complete web browsers, it becomes possible to use web techniques to trigger the authentication process in a simple, standards-compliant way. Authentication may then be achieved by bridging simple web protocols with the device's existing authentication mechanism. In addition, because the user typically keeps her mobile device with her at all times, it may be usable as an authentication token for desktop-browser access, again using existing, standard web protocols.

The Path: a Standard JavaScript Authentication API. To be deployable and usable as widely as possible, a web-based authentication mechanism must be easily triggered by a web site. We suggest the creation of a standard API, in JavaScript, that would allow web sites to begin a client-side authentication process. Such an API, e.g. a `window.auth` library, would be easily discoverable, much like certain advanced browser-specific JavaScript features are used on an as-available basis. For example, a web site might issue the following (strawman) JavaScript code:

```
if (window.auth) {
    token = window.auth.requestAuth(
        {/* fields to authenticate */}
    );
} else {/* revert to weak password authentication */}
```

The specifics of the API need to be carefully thought out to enable:

1. multiple types of authentication,
2. discovery of supported authentication methods, and
3. evolution of authentication methods over time.

In any case, a call to `window.auth.*` should cause a clearly recognizable login ritual, un-spoofable by a typical web page, where the user may be required to use a specific physical button to unlock the authentication token.

Conclusion. It has long been recognized that mobile devices offer a particularly interesting opportunity to provide authentication for other channels: mobile devices are already tied to long-lasting user accounts, and the hardware evolves quickly enough that new, more secure UI paradigms can be explored. We propose to take a standards-based web approach to this problem: any web site should be able to trigger a strong authentication process using simple JavaScript. The underlying specifics of the login ritual can be device- and carrier-specific.

As mobile devices truly become trusted user proxies for digital transactions, it will be particularly interesting to web-enable authentication and authorization processes in general, so that even financial and medical transactions may be achieved via web-based applications.

References

[1] Kim Cameron. As simple as possible – but no simpler. http://www.identityblog.com/?p=649, last visited on February 3rd 2007.

[2] Kim Cameron and Michael B. Jones. Design Rationale behind the Identity Metasystem Architecture, 2006. http://www.identityblog.com/wp-content/resources/design_rationale.pdf.

[3] Markus Jakobsson and Steven Myers. *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft.* Wiley-Interscience, 2006.

[4] Ben Laurie. OpenID: Phishing Heaven. http://www.links.org/?p=187, last visited on February 3rd 2007.

[5] D. Recordon and B. Fitzpatrick. OpenID Authentication 1.1, May 2006. http://openid.net/specs/openid-authentication-1_1.html.

[6] RSA Data Security. RSA Mobile Authentication Solution, Sep 2002. http://www.rsa.com/press_release.aspx?id=1370.

[7] U Sax, I Kohane, and K D Mandl. Wireless technology infrastructures for authentication of patients: PKI that rings. *J Am Med Inform Assoc*, 12(3):263–268, May-Jun 2005.

[8] Min Wu, Simson L. Garfinkel, and Robert Miller. Secure web authentication with mobile phones, 2005. http://groups.csail.mit.edu/uid/projects/cellphone-auth/.