Thoughts on Trust Infrastructure, User Interface, and Legal Issues for the W3C "Identity in the Browser" Workshop, May 24-25, 2011

Stephen Schultze
Associate Director
Center for Information Technology Policy
Princeton University

Thomas Lowenthal
Princeton University

The topic of this workshop is particularly timely, as a variety of factors have converged to make identity in the browser especially relevant at this moment. We have witnessed recent disturbing failures of our existing trust infrastructure, significant changes in the identity user interface of desktop and mobile browsers, and a renewed interest in understanding the true legal structure and liabilities behind the identity technologies we rely on every day. I will briefly address each of these in turn, drawing from some of my preliminary writing on the matter. I hope that this can serve as a prompt for more complete discussion amongst the workshop participants.

Trust Infrastructure[1]

Web browsers currently use a *flat* and *inflexible* trust model. Each browser or operating system comes pre-loaded with a list of Certificate Authorities that it will trust to guarantee the authenticity of web sites you visit. Each vendor makes its own list.[2] This is a *flat* model because each CA has just as much authority as the others, thus each effectively sits at the "root" of authority. Indeed any of the CAs can sign certificates for any entity in the world. They do not coordinate with each other, and can sign a certificate for an entity even if another CA has already done so. This weakness was recently highlighted when an Iranian attacker compromised Comodo's system and issues certificates to himself for several major web sites.[3] Furthermore, certificate authorities can confer this god-like power on other entities without oversight or the prior knowledge of the end users or the entities being signed for. This is also an *inflexible* model because there is no reasonable way to impose finer-grained control on the authority of the CAs. Browsers and X.509 do not allow you to trust Verisign to a greater or less than a potentially hostile government—it is essentially all or nothing for each. Users also can't tell their browser to trust particular CAs for particular TLDs (although domain name constraints do exist in the standard, they are not widely implemented). It is also inflexible because most browsers intentionally make it difficult for a user to change the certificate list. It might be possible to partially mitigate some of the CA/X.509 shortcomings by implementing more constraints, improving the user interface, adding "out of band" certificate checks,[4] or generating more paranoid certificate warnings.[5]

---

[1] Adapted from my blog post here: http://www.freedom-to-tinker.com/blog/sjs/web-security-trust-models
[2] See for example Mozilla's Certificate Policy for the standards for being added to this list: http://www.mozilla.org/projects/security/certs/policy/
[3] http://freedom-to-tinker.com/blog/sjs/web-browsers-and-comodo-disclose-successful-certificate-authority-attack-perhaps-iran
[4] eg: http://www.networknotary.org/
[5] eg: http://patrol.psyced.org/

However, the most promising approach the improving the current trust infrastructure is to transition to something more *hierarchical and delegated*. This approach starts with a single highly trusted root and delegates authority recursively.[6] Any authority can only issue certificates for itself or the entities that fall "underneath" it, thus limiting the god-like power of the flat model. This also pushes signing power closer to the authenticated sites themselves. It is possible that this authority could be placed directly in their hands, rather than requiring an external authority to approve of each new certificate or domain.

I am describing this in a very domain-centric way. If we are willing to fully buy into the domain hierarchy way of thinking about web security, there may be a viable implementation path for this model. Perhaps the greatest example of this delegation approach to web governance is the existing Domain Name System. Decisions at the root of DNS are governed by the international non-profit ICANN, which assigns authority to Top Level Domains (eg: .com, .net, .cn) who then further delegate through a system of registrars. Historically, the biggest problem with tying site authentication to DNS has been that that DNS is deeply insecure. However, a more secure version of DNS, DNSSEC, has now been deployed at the DNS root and implemented by most of the major TLDs.[7] Any DNSSEC query can be verified by following the chain of authority back to the root, and any contents of the response can be guaranteed to be unaltered through that chain of trust. This infrastructure could be the basis for distributing site certificates as well, which could form the basis for hierarchical site authenticity checking.

Work on this approach is at an advanced stage, with a standards-track IETF working group[8] and a Firefox plugin proof-of-concept.[9]

If implemented on a wide scale, this approach would thoroughly bind web site authentication to the DNS hierarchy, and the only assurance it would provide is that you are communicating with the person who registered the domain you are visiting. It would not provide any additional verification about who that person is, as Certificate Authorities theoretically could do (but practically don't). Certificates were originally envisioned as a way to guarantee that a particular real-world entity was behind the site in question, but market pressures caused CAs cut corners on the verification process. Most CAs now offer "Domain Validation" (DV) certificates that are issued without any human intervention and simply verify that the person requesting the certificate has control of the domain in question. These certificates are treated no differently than more rigorously verified certificates, so for all intents and purposes the DNSSEC certificate delegation model would provide at least the services of the current CA model. One exception is Extended Validation certificates, which require the CA to perform more rigorous checks and cause the browser URL bar to take on a "green glow".

---

[6] See Dan Kaminsky's version of this approach, *Introducing The Domain Key Infrastructure*: http://www.slideshare.net/dakami/domain-key-infrastructure-from-black-hat-usa

[7] http://www.freedom-to-tinker.com/blog/sjs/major-internet-milestone-dnssec-and-ssl

[8] http://tools.ietf.org/html/draft-ietf-dane-protocol

[9] https://os3sec.org/

User Interface[10]

The most ubiquitous indicator of a "secure" connection on the web is the "padlock icon." For years, banks, commerce sites, and geek grandchildren have been telling people to "look for the lock." However, The padlock has problems. First, it has been shown in user studies that despite all of the imploring, many people just don't pay attention.[11] Second, when they do pay attention, the padlock often gives them the impression that the site they are connecting to is the real-world person or company that the site claims to be (in reality, it usually just means that the connection is encrypted to "somebody"). Even more generally, many people think that the padlock means that they are "safe" to do whatever they wish on the site without risk. "Extended Validation" (EV) certificates provide an interesting tweak to this model. As opposed to "Domain Validation" (DV) certs that simply verify that you are talking to "somebody" who owns the domain, EV certificates actually do verify real-world identities. They also typically cause some prominent part of the browser to turn green and show the real-world entity's name and location (eg: "Bank of America Corporation (US)"). Separately, the W3C recently issued a final draft of a document entitled "Web Security Context: User Interface Guidelines."[12] The document describes web site "identity signals," saying that the browser must "make information about the identity of the Web site that a user interacts with available." These developments highlight a shift in browser security UI from simply showing a binary padlock/no-padlock icon to showing more rich information about identity (when it exists). However, the W3C guidelines don't seem to indicate much of anything specific.

In the context of growing importance of mobile browsing and increasing competition on user interface, all of the browsers have been making changes to the security/identity user interface. All of the browsers other than Firefox still have a padlock icon (albeit in different places). Chrome now makes "https" and the padlock icon green regardless of whether it is DV or EV,[13] whereas the other browsers reserve the green color for EV only. The confusion is made worse by the fact that Chrome appears to contain a bug in which the organization name/location (the only indication of EV validation) sometimes does not appear. Firefox chose to use the color blue for DV even though one of their user experience guys noted, "The color blue unfortunately carries no meaning or really any form of positive/negative connotation (this was intentional and the rational[e] is rather complex)".[14] The name/location from EV certificates appear in different places, and the method of coloring elements also varies (Safari in particular colors only the text, and does so in dark shades that can sometimes be hard to discern from black). Some browsers also make (different) portions of the URL a shade of gray in an attempt to emphasize the domain you are visiting. Mozilla has been particularly aggressively changing Firefox's user interface, with the most dramatic change being the removal of the padlock icon entirely as of Firefox 4.

---

[10] Adapted in part from my more detailed post that includes screenshots, here: http://freedom-to-tinker.com/blog/sjs/web-browser-security-user-interfaces-hard-get-right-and-increasingly-inconsistent

[11] http://www.usablesecurity.org/emperor/

[12] http://www.w3.org/TR/wsc-ui/

[13] See debate here: https://code.google.com/p/chromium/issues/detail?id=41481#c12

[14] https://bug588270.bugzilla.mozilla.org/attachment.cgi?id=466899

Are these changes a good thing? On the one hand, movement toward a more accurately descriptive system is generally laudable. On the other, it seems disturbing that the browsers are diverging in their visual language of security. I have heard people argue that competition in security UI could be a good thing, but I am not convinced that any benefits would outweigh the cost of confusing users. I'm also not sure that users are aware enough of the differences that they will consider it when selecting a browser—limiting the positive effects of any competition. What's more, the problem is only set to get worse as more and more browsing takes place on mobile devices that are inherently constrained in what they can cram on the screen. There nevertheless seems to be an opportunity here for some standardization amongst the browser vendors, with a foundation in actual usability testing. I would urge a renewed effort on the part of browser vendors to work together to implement consistent and empirically tested interfaces.

Legal Issues[15]

The weaknesses of the current browser trust model and UI identity indicators are compounded by problems with the legal documentation typically associated with the model. CAs often employ several standard documents.

The *subscriber agreement* is a contract between the CA and the owner/operator of the domain name. It sets forth the terms and conditions governing the CA's issuance of SSL certificates to the Web site operator and the operator's subsequent permitted use of those certificates.

The *certification practice statement* (CPS) is a separate document that describes the business practices regarding the CA's issuance of digital certificates. The CPS purports to limit the CA's monetary liability and limit the extent to which a subscriber or relying party (in the latter case, an end-user with a browser) may rely on authentication or encryption methods that use the CA's certificates. The terms of the CPS are typically incorporated by reference in the subscriber agreement, but there appears to be no mechanism by which they are presented to the relying party for his or her approval or assent. According to typical language included in the CPS, one might readily conclude that CAs do not stand behind the digital certificates that they issue:

> Warranties and Limitations on Warranties
> In no event does [the CA] ... make any representations, or provide any warranties ... to any ... Subscribers, Relying Parties, or any other persons, entities, or organizations with respect to ... the reliability of any cryptographic techniques or methods used in conducting any act, transaction, or process involving or utilizing [a] Certificate ....[16]

---

[15] Adapted in part from an article I co-authored with Steven Roosa: http://www.freedom-to-tinker.com/blog/sroosa/flawed-legal-architecture-certificate-authority-trust-model

[16] http://www.entrust.net/CPS/pdf/SSL-CPS-English-160810-v2-4.pdf

The third document, the *relying party agreement*, purports to be an agreement between the CA and the relying party/end-user. This document often purports to place onerous technical obligations on the end-user, such as being familiar with the underlying cryptographic protocols and making independent judgments about the trustworthiness of any given digital certificate. A typical relying party agreement also contains a significant liability disclaimer (by the CAs) to end-users for defects in authentication. The end-user's assent to these standard documents is generally neither obtained nor sought. There appears to be no occasion when an end-user clicks his or her assent to the relying party agreement, the CPS, or the subscriber agreement. As far as the end-user is concerned, these documents do not exist.

The absence of assent by the end-user places the Web site operator that is a "subscriber" to the CA's SSL certificates in a difficult position, as Web site operators are actively encouraging end-users to rely heavily on SSL encrypted communications, while entering into contracts with CAs that seek to minimize, if not eliminate, the end-user's right to rely on the authentication processes on which SSL communications depend. A review of the published decisional law fails to reveal any court decision that speaks directly to the issue of end-user rights relative to the legal documentation associated with the CA Trust Model. As a result, the legal architecture on which the model rests is untested.

Any comprehensive effort to improve the state of identity in the browser must consider the underlying legal infrastructure implied in technical decisions. The original ABA Guidelines anticipated several of these issues, but did not provide conclusive answers and have not been updated over the years. Other efforts, such as the National Strategy on Trusted Identities in Cyberspace[17] and the ABA's own Federated Identity Management Legal Task Force[18] have explored these topics at a high level, but have yet to produce conclusive and workable legal regimes and trust models.

Conclusion

In my opinion, the task of improving identity in the browser has tended to yield several unhelpful or incomplete outcomes: grand schemes for federated identity that are dead on arrival, specific individual technical interventions that remain un-implemented, and inconsistencies across browsers that hinder user awareness and adoption. These are a result of failures in the three domains that I described, and more. I hope that this workshop can be a platform for overcoming some of these challenges.

[17] http://www.nist.gov/nstic/
[18] http://apps.americanbar.org/dch/committee.cfm?com=CL320041