Building Secure Mashups

D. K. Smetters
PARC
3333 Coyote Hill Rd, Palo Alto CA 94304
smetters@parc.com

Your data – anywhere, anytime, combined with that of your friends (or anyone else's you can get your hands on) in any way you might think interesting – that is the promise of Web 2.0. Achieving this vision, however requires the ability to build mashups, or *data transformation services* – that operate on any or all of the following sources of data:

- User-generated data – often *personal* user-generated data, such as photographs
- Social network information – another form of private, and even valuable data
- Public or semi-public data sources – databases of available information (e.g. Google Maps) with varying guarantees of correctness and constraints on use
- Private data sources – e.g. corporate data subject to some form of access control

The result of such a data transformation service is either a user-focused result (e.g. a visualization), or a derived data source that can feed into further mashups. In its most general form, anything that combines two different sources of data in a useful way could be considered such a mashup, and the provider of the data transformation service need not have any privileged relationship with the holder of the data. It is the data owner, not the site holding the data, who must decide where that data can or should go. In essence, exploiting the full promise of Web 2.0 requires building systems designed for flexible, ad-hoc cross-organization delegation of limited access to sensitive data – *all under easy user control*.

We argue in this position paper that although recent technological developments have begun to provide the machinery to make secure mashups possible, they have underestimated the challenges involved in making that security usable. By considering the requirements posed by potentially multi-step secure mashups in the context of the security and usability failures of today's much simpler web applications, we can see how hard this problem will be.

To date, mashups have addressed this problem in one of five ways:

- **Avoidance:** Use only public or semi-public data (e.g. data whose use is throttled to prevent denial of service attacks). This avoids the problem of access to sensitive data entirely.
- **Reduce it to a previously solved problem:** The service operating on the sensitive data is the one that already holds the data, or one with whom the user has an existing trust relationship sufficient to allow blanket delegations (e.g. handing over your username and password for one data provider to another to allow the second provider to import data from the first.) This allows access to sensitive data, but not in a manner that is either flexible, ad-hoc (extensible to interactions with new or untrusted services) or limited.

- **Looking under the lamppost:** For some applications, it is possible to identify simple collections of data that the user might want to delegate access to *a priori*, and make available easy means of delegating such access (e.g. social networking sites that allow users to grant access to their list of friends to otherwise untrusted applications via standard APIs and tickets, tokens, or cookie- or session-based identifiers). This allows limited and ad-hoc access, but is not particularly flexible; addressing only access to predefined collections of data.

- **Building a bridge:** Particularly in corporate mashups (i.e. service-oriented architectures), business relationships, special access credentials or accounts, or even special servers or protocols can be set up to effectively tailor access to particular data by a designated service. The classic forms of this approach are dedicated financial data aggregators (Bloomberg), or B2B networks. This is limited, but neither flexible or ad-hoc.

- **Outsourcing:** In order to make limited delegation of access possible while protecting user password-based credentials, the user maintains a relationship with an identity provider (e.g. OpenID, Cardspace) or an authorization service (e.g. OAuth, a SAML server); the former allow the user to federate their own identities across multiple service providers and supports movement of data between them, the latter allows the user to interact with a trusted provider that mediates cross-domain delegation of access to protected resources. This provides a technical foundation for flexible cross-domain delegation, but leaves open the challenge of connecting these mechanisms to the user.

Connecting with the User

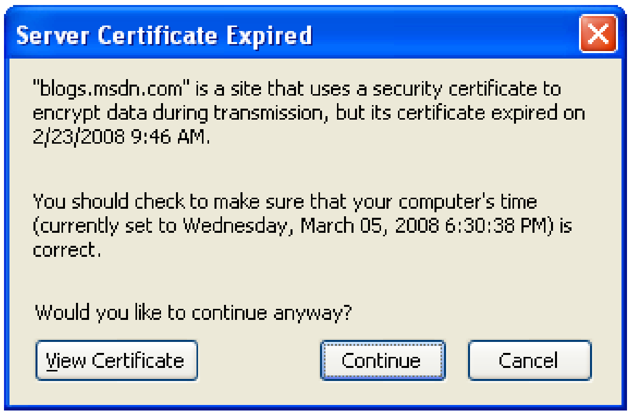
Ellison [1] coined the use of the term *ceremony* to refer to the larger context of a security application or protocol, one which the user's thoughts and actions are included in the analysis. When we consider the ceremonies involved in secure mashups, we can identify three communities of users: mashup developers, administrators and owners of data to be mashed, and the end users of the resulting systems; and a list of problems such a system must enable those users to solve:

1. **Connecting the providers they mean to connect:** The success of phishing attacks has clearly demonstrated that current approaches to allowing users to authenticate hosts, services, or other Internet entities have failed [3,4]. While TLS and digital certificates are good mechanisms for allowing hosts to know "who" it is they are securely communicating with, it is a much greater challenge to determine whether that digital identity matches the user's intent. In the mashup context, this means that even if you outsource delegation to an authentication provider (and get the user to sign up with one, etc, etc), you will have difficulty helping the user ensure they are actually talking to the provider they intend – and that is when that interaction is "face to face" in a web browser, not to mention when it is buried in a pipeline of connected services. (See figure 1, below).

2. **Specifying policies:** While the most interesting untapped source of mashup data is that protected inside corporate firewalls, that data is also subject to the clearest policies and the largest resources for specifying those policies. It is the multitude of personal data belonging to their friends and associates that people would like to mash that pose the greater challenge. Studies have shown that the policies people apply, or would like to apply, to their personal data can be complex [4,5], and that current simplified user interfaces for specifying such policies often do not capture users' needs [6]. While work has been done on generating requirements for

such policies [7], getting users to specify them, or cope with exceptions and processing in real time is largely an open problem (see [8,9] for alternate approaches to the problem).

3. Identifying data: The ability to uniquely refer to a user-sensible collection of data is a powerful tool in effectively delegating access. For example, the successful approaches to limited delegation in social networking sites rely on the fact that the information to be made available is clear to the user ("my list of friends"), and the number of such collections is small enough for the user to manage. More general forms of mashups require users to be able to reason about, manage policies for, and potentially answer questions about (Should site xyz have access to the "Other Photos" folder?) much more flexible collections of data for which no *a priori* policies can be automatically derived on the user's behalf. Not only must the user remember what he put in the "Other Photos" folder in this example, but he must be confident that both the data holder, the service provider, and any intermediaries helping glue together the security on his behalf mean the same thing.


Server Certificate Expired

"blogs.msdn.com" is a site that uses a security certificate to encrypt data during transmission, but its certificate expired on 2/23/2008 9:46 AM.

You should check to make sure that your computer's time (currently set to Wednesday, March 05, 2008 6:30:38 PM) is correct.

Would you like to continue anyway?

Figure 1. Dialogs that should not happen.

4. Making the user an ally: If the user is to have control over their information, they must be an active participant in determining where it goes, and what happens to it. That means the system must enforce the policies it says it does, and it must be possible to determine when something has gone wrong. While this sounds simple enough, consider the following: a user receives a security warning, telling her that her current action is subject to attack. She needs to perform that action to get her job done. Does she heed the warning? Only if she truly believes that it might be correct. When would she believe that it might be correct? When the number of times she has gotten such a warning in error – for example, due to a system misconfiguration rather than an actual attack – is vanishingly small. Unfortunately, even with the relatively simple and largely ineffective security measures in place on the internet today, the frequency of configuration errors is so high (as of Jan. 1, 2008, 68% of web server SSL(TLS) certificates were invalid [10]). If users are to do anything at all to aid in their own defense, not only must the information they receive be in terms they can understand, but it must be correct. This is especially true in the context of a mashup, where a security error may be detected deep in a pipeline of connected services, with a human far down at one end. For a simple example, consider the dialog box in Figure 1, which appears at random intervals for one user because the server supplying the RSS feed he has embedded in his browser happened to choose to authenticate its data, and then let its server certificate expire.

Conclusions

In this paper, we have argued that current and upcoming mechanisms for securing ad-hoc connections between data and processing services – mashups – have yet to consider the ceremonies involved in using and managing such services, and bring the user into the security picture. We have identified four major classes of user-focused problems such services must address. Solving these problems should be a key goal of future work on Web 2.0 security.

References

1. Ellison, C., *Ceremony Design and Analysis*, Cryptology ePrint Archive, Report 2007/399, 2007, http://eprint.iacr.org/2007/399.

2. Dhamija, R., Tygar, J. D., and Hearst, M. 2006. *Why phishing works.* In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM, New York, NY, 581-590. DOI= http://doi.acm.org/10.1145/1124772.1124861

3. Schechter, S. E, Dhamija, R., Ozment, A., and Fischer, I.. 2007. *The Emperor's New Security Indicators.* Washington, DC, USA : IEEE Computer Society, 2007. SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy. pp. 51-65.

4. Miller, A.D. and Edwards, W.E. *Give and take: a study of consumer photo-sharing culture and practice*. In CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 347–356, New York, NY, USA, 2007.

5. Olson, J.S., Grudin, J. and Horvitz, E.. *A study of preferences for sharing and privacy*. In CHI '05: CHI '05 extended abstracts on Human factors in computing systems, pages 1985–1988, New York, NY, USA, 2005.

6. Ahern, S., Eckles, D., Good, N.S., King, S., Naaman, M. and Nair, R. *Over-exposed?: Privacy Patterns and Considerations in Online and Mobile Photo Sharing*. In CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems 357–366, New York, NY, USA, 2007.

7. Gates, C.. *Access Control Requirements for Web 2.0 Security and Privacy*. Web 2.0 Security and Privacy Workshop, 2007. http://seclab.cs.rice.edu/w2sp/2007/papers/paper-205-z_708.pdf

8. Bauer, L., Garriss, S., McCune, J., Reiter, M., Rouse, J., and Rutenbar, P.. *Device-enabled authorization in the Grey system*. In Proceedings of the 8th Information Security Conference (ISC'05), pages 431--445, Singapore, September 2005. Springer Verlag LNCS 3650.

9. Smetters, D. K.; Balfanz, D.; Durfee, G. E.; Smith, T.; Lee, K*. Instant matchmaking: simple, secure virtual extensions to ubiquitous computing environments*. Ubicomp 2006, Proceedings of the 8th International Conference of Ubiquitous Computing; 2006 September 17-21; Irvine; CA; USA. Berlin: Springer Verlag; 2006; LCS 4206: 477-494.

10. SecuritySpace. 2008. *Secure Server Survey*. SecuritySpace. [Online] January 1, 2008. http://www.securityspace.com/s_survey/sdata/200712/certca.html.