Client Authentication in Federations Using a Security Mode

Sebastian Gajek and Ahmad-Reza Sadeghi

Horst Görtz Institute for IT-Security
Ruhr University Bochum, Germany

Abstract

Nowadays, identity-based client authentication (e.g., by username/ password) over SSL is the standard for user authentication on the Web. In particular, browser-based federated identity management (FIM) protocols prefer this technique to authenticate customers due to its user-convenience and lightweight access management. However, recent attacks known as phishing provide evidence that this authentication scheme is vulnerable to identity theft. As a consequence, FIM-protocols are likewise threatened by online-fraud. More dramatically, since FIM-protocols grant access to a federation of services with solely a single identity, a misuse affects many services equally. Therefore, we propose to define a secure mode (FIM-M), which downgrades malicious features of web browsers in the case of FIM and let us more concisely reason about the security of FIM-protocols.

1 Background

Single sign-on approaches aim to offering one-click features across the entire world-wide web [5]. Ideally, a password and some clicks should transfer both the user's identity and relevant user information (also called attributes as generalizing term), which were deposited at a trusted third party acting as credential supplier. The main goal is to provide a user with a single identity (federated identity) for ease-of-use that she can reliably reuse in a federation of trusted services. Several proposals for *federated identity management* (FIM) protocols have been made so far. Examples include the Shibboleth project [1], Liberty Single Sign On [11], .NET passport [12], SAML [2], or WS-Federation [4]. The chief requirement on above mentioned FIM-protocols is that a standard web browser should be able to provide a federated identity. An additional aggravation is that the user should be able to access her federated identity independently of her location, such as at Internet kiosks. Hence, all one can rely on is a web browser supporting standard web languages, and in particular transport protocols, such as HTTP triggered over SSL.

Recent studies point out (e.g., [3]) that such a setting is susceptible to so-called phishing attacks. Adversaries imitate the look and feel of legitimate web sites including tampering with SSL security indicators to lure ordinary users to counterfeit web sites and suggest users to disclose their identity. By the day, destined services (e.g., online-banking) are threatened by these attacks. In mid-term, we expect that phishing attacks will also proliferate and affect FIM-services:

- The idea of federated identities was originally to advance electronic commerce (ecommerce) and thus tailored to a wide range of consumers. Having our lessons learned, we must not presuppose that most of these consumers are highly security-diligent. In our opinion, the use of SSL in FIM-protocols is the only cryptographic mechanism we can rely on, if we consider a minimal setting, i.e. the user is equipped with a standard web brower solely. Unfortunately, ordinary users—the peak of consumers we consider—are unable to trustworthily identify an SSL-protected connection (e.g., all bogus phishing sites might have been disclosed, if a user had properly verified the server certificate). Consequently, we must assume that stealing a federated identity is feasible with means, phishing attacks currently use.

- Identity theft on the Internet is linked to misuse (more precisely, to money laundering) and mostly yields financial losses of a betrayed customer. Therefore, financial services are lucrative targets. However, as the attack's nuisance increases both in volume and sophistication, adversaries will focus on emerging new markets. We believe that phishing attacks will unfurl in ecommerce services as commerce of today provides a variegated portfolio of services offering phishers a plethora of new applications (e.g., Pay-per-Download, Voice-over-IP) indirectly leading to comparable monetary enrichment. Unfortunately under this circumstances, a compromised federated identity equals a complimentary ticket to a federation of services.

Based on this, we conclude that by today FIM does not achieve in practice a sufficient level of security to protect users from misuse of their identity, although recent work [8, 7] attempted to formally analyze some FIM-protocols in security models (to prove security properties, such as authentication under certain assumptions). Hence, a weak level of security is crucial for a long-term establishment of federated identities and it is mandatory to update future releases of web browsers; it is essential to discuss new security features filling the gap to secure the technology of federated identity management.

2 Our proposal

Several browser-based approaches have been made to counteract phishing attacks. However, these approaches are tangential to our proposal. Since the ideas

deter phishing attacks, they also counter federated identity theft. Nevertheless, regarding protection of FIM-services, we have to bear in mind their specific requirements:

Usability A key requirement is ease-of-use. Users may not be distracted from any sophisticated interaction. Therefore, the user interface of a web browser is an essential part in browser-based FIM-protocols. Moreover, it is important that a web browser relaxes the user's burdens. Ideally, browser and server should reclusively interact to establish a trusted channel, i.e. the user must be relaxed to verify the security properties. Unfortunately, a web browser is unaware of the higher protocol it is involved in: In common security protocols, principals are assumed to execute precisely the security protocol under consideration unless they are corrupted. A browser, in contrast, reacts on a set of predefined messages, adds information to responses *automatically*, and stores certain information such as histories in places, which cannot always be assumed to be secure. This is a security concern having direct impact on the security of the higher protocol (see, e.g., [6] for a concrete real world attack scenario on SAML.). We call such a party (like the browser) *protocol-unaware*.

Restrictivity FIM provides access to a federation. These federations are closed groups of services, which share a trust relationship according to security policies. A major benefit in comparison to unique services is that we are able to stipulate a federated security policy. As we define how services communicate among each others, how they negotiate and access credentials, we can prerequisite how services are presented. Note that this is a complete new situation. A widely accepted position is that the limitation of modern web languages (e.g., JavaScript, ActiveX) narrows the attraction of the Internet. We fully agree with this opinion. Indeed, it is not practicable to reduce all features of modern web design in a general case. However, in a closed group we can more easily enforce that services follow common design principles, as we are able to define federated security policies.

We propose to trigger the browser into a special mode (FIM-M) that tames the browser into a high security mode and follows both security guidelines. This is mandatory because the browser is protocol-unaware. Solely if we tame the browser we can expect how a FIM-protocol adheres. In other words, if we anticipate the states of a web browser, we are able to point out a sound protocol run and reason about the security. As mentioned before in, e.g., [8, 7] the authors already elaborated a model to analyze browser-based protocols.

In detail, FIM-M limits features, which can be potentially misapplied to trick the user about a web site's authenticity.[1] Concretely, we propose to reduce the browser to the notion of "zero-footprint", i.e. the browser supports solely rudimentary web language features. We do not claim that "zero-footprint"

[1] Note that this requirement does not protect against any malware attack. This is out of our scope.

solves completely the problem of identity theft, but rather domesticates adversaries. Our goal is to prohibit the fakeability of security-relevant parts of the user-interface. A moderately educated user should always be able to see properly displayed security and connection information. Today, web browsers are augmented with masses of features that obfuscate the real presentation. For instance, an adversary is still able to remotely turn off the address bar (displaying the domain name—one fundamental indicator of a web site's authenticity). Finally, we must return to the point that users "see what they get". Second, we must ensure that users are able to understand SSL. However, this is an awkward task, as a "non-cryptographer" ought to understand how to properly indicate a trustworthy SSL-connection. We believe that today's web browser do not provide the required convenience of non-cryptographers; vice versa, we have to accept that users (or a notable amount of them) will never provide the required diligence of cryptographers; security is made for laymen, and not for those who are experts in protecting themselves. Therefore, we encourage work that deploys non-cryptographic means to display security parameters. In this line, we see fruitable examples, such as Trustbar [9] or DPA [10], which deploy visual representations of cryptography.

A challenge we face is how to activate this mode. We prefer a remote activation to again relax the user' s burdens. Ideally, this can be achieved, when browsers would automatically detect FIM-protocols (e.g., by recognizing a special HTML tag). This is not a crucial task as we wrap the browser into a secure mode and tame his functionality. An adversary would rather intend to find means how to circumvent this mode, i.e. to gain access to features laying the grounds of his attack. Today, users can manually realize some rudimentary aspects of FIM-M. For instance, the IE6 provides to configure Internet zones, which set up security levels. However, this feature is rarely used because (a) a user must take care of the configuration and (b) must verify, if indeed the level has been activated. Vice versa, a remote activation has the advantage that the FIM-provider could ensure a client's browser is securely wrapped.

3 Summary and Future Work

In a discussion we would like to contribute two parts.

- First, we outline the peril current web authentication mechanisms pose on federated identity management. For the sake of briefness, we focus our discussion on browser-based protocols and pinpoint flaws, which were found in the past and outline features that in particular threat the security of FIM-protocols.
- Second, we sketch a wish list of features and non-features we expect in future releases of web browsers to fulfill a high level of security. We see the web browser as the most important responsibility for the wide-spread adaption of FIM. As the user's interface to a federation of services, security features of a web browser mainly will impact trust in federated identities.

References

[1] *Shibboleth-Architecture Draft v05*, May 2002 (v01 in 2001). http://middleware.internet2.edu/shibboleth/docs/draft-internet2-shibboleth-arch-v05.pdf.

[2] *Security Assertion Markup Language (SAML); OASIS Standard*, November 2002. http://www.oasis-open.org/committees/security/docs/.

[3] A. Adelsbach, S. Gajek, and J. Schwenk. Visual Spoofing of SSL Protected Web Sites and Effective Countermeasures. In *Information Security Practice and Experience Conference*, 2005.

[4] BEA, IBM, Microsoft, RSA Security, and VeriSign. *WS-Federation: Passive Requestor Profile; Draft, Version 1.0.* http://www-106.ibm.com/developerworks/webservices/.

[5] Birgit Pfitzmann and Michael Waidner. BBAE A General Protocol for Browser-based Attribute. Technical report, IBM Zürich, 2002.

[6] T. Groß. Security analysis of the saml single sign-on browser/artifact profile. In *19th Annual Computer Security Applications Conference (ACSAC 2003)*. IEEE Computer Society Press, 2003.

[7] T. Groß, B. Pfitzmann, and A.-R. Sadeghi. Proving a ws-federation passive requestor profile with a browser model. In *ACM Workshop on Secure Web Services (SWS)*, pages 54–64. ACM Press, November 2005.

[8] T. Groß, B. Pfitzmann, and A.-R. Sadeghi. Browser model for security analysis of browser-based protocols. In *10th European Symposium on Research in Computer Security (ESORICS)*, volume 3679 of *Lecture Notes on Computer Science*, pages 489–508. Springer Verlag, September 2005. Earlier version IBM Research Report RZ 3600 (#99610).

[9] A. Herzberg and A. Gbara. TrustBar: Protecting (even Naive) Web Users from Spoofing and Phishing Attacks. IACR Cryptology ePrint Archive, 2004.

[10] M. Jakobsson and S. Myers. Stealth Attacks and Delayed Password Disclosure., 2005.

[11] Liberty Alliance Project. *Liberty Phase 2 Final Specifications*, November 2003. http://www.projectliberty.org/specs/lap-phase2-final.zip (v1.0 July 2002).

[12] Microsoft Corporation. *.NET Passport documentation, in particular Technical Overview*, September 2001. http://www.passport.com and http://msdn.microsoft.com/downloads.