# BROWSER SUPPORT FOR THE OPEN AUTHORIZATION (OAUTH) PROTOCOL

Hannes Tschofenig, Barry Leiba, Blaine Cook, Rob van Eijk

## I. INTRODUCTION

The Open Authorization (OAuth) protocol [1] allows a user to grant a third-party Web site or application access to the user's protected resources, without necessarily revealing their long-term credentials, or even their identity. For example, a photo-sharing site that supports OAuth could allow its users to use a third-party printing Web site to print their private pictures, without allowing the printing site to gain full control of the user's account.

OAuth is a fairly flexible protocol that can be deployed by third party websites as well as by downloadable applications on end devices. The currently ongoing work in the IETF OAuth working group [2] aims to standardize the core components, while other parts are left for further work (such as token encoding and token content) or are outside the scope of IETF standardization (such as user-interface specifications).

With this position paper the authors would like to highlight the need to enhance the functionality of browsers for better support of OAuth.

## II. BROWSER ENHANCEMENTS

We believe that the following enhancements to the browser would be beneficial for more secure OAuth deployments, as well as for more secure deployment of other identity and authorization frameworks – some of the features we describe will be common and re-usable.

### *A. Authentication Mechanisms*

Part of the OAuth protocol exchange requires the user to be redirected from the third party website back to the protected resource for authorization. For an authorization dialog to be presented to the user, an authentication exchange needs to have taken place.

Strictly speaking, the exchange to authenticate the user is outside the realm of standardization for OAuth (as is also the case for many other identity management solutions, such as OpenID and the SAML Web SSO profile). For the security of the overall system, however, it is of critical importance that this step be performed in a secure way. While one part of the overall security quality is related to the type of credential being used, the initial enrollment step is also important. Once credentials have been enrolled and are available they need to be used by a specific authentication and key exchange protocol. The variety of protocols available for this purpose reflects the different requirements found in various environments. Clearly, there is no one-size-fits-all authentication and key exchange protocol.

*This position paper is a submission to the W3C Workshop on Identity in the Browser, 24/25th May 2011, Mountain View (USA).

HANNES TSCHOFENIG is a senior standardization specialist at Nokia Siemens Networks. He is active at the Internet Engineering Task Force (IETF), co-chair of the Open Authentication Protocol (OAuth) working group, and a member of the Internet Architecture Board (IAB). He can be reached at hannes.tschofenig@nsn.com.

BARRY LEIBA is a Standards Manager at Huawei Technologies. He co-chairs five IETF working groups, including OAuth and DKIM, is active in the security directorate, and served on the Internet Architecture Board from 2007 to 2009. He can be reached at barryleiba@computer.org.

BLAINE COOK is a principal co-author of the original OAuth 1.0 specification and former lead developer of Twitter. He is now with British Telecom and co-chairs the OAuth working group. He can be reached at romeda@gmail.com.

ROB VAN EIJK joined the Dutch Data Protection Authority (Dutch DPA) in April 2010 where he works as an Internet Technology Expert. Prior to joining the Dutch DPA, he conducted IT Forensics and Fraud Investigations as a contractor. Currently, he is a PhD candidate in Computer Science at Leiden University. He holds an APMG approved PRINCE2® Lead Trainer qualification in project management and has been a visiting lecturer in agile project management at the Software School of Xiamen University, and has been contributing to the China Holland Educational Competence and Knowledge Center on IT (CHECK-IT). He can be contacted at r.j.van.eijk@umail.leidenuniv.nl.

Without recommending specific authentication and key exchange protocols or suggesting different types of credentials to be used, it's very important that a discussion takes place about how to improve the current Web browser support for these security features.

Relevant work includes WebPKI [3], PSCK [4], and the ongoing work in the ABFAB working group [5], [6].

### *B. Authorization Interface*

Once the authentication exchange described in the previous item is completed, a subsequent step is to present the user with an authorization dialog. According to good privacy design, the user has to be informed about the information he or she is about to share with another party, for what purpose, and under what conditions (e.g. retention period).

Unfortunately, current practice shows that these dialogs are often misleading and do not provide the needed minimum information to enable the user to make an informed decision. See [7] for an initial outline of these concerns.

Clearly, user interface design is difficult but the current situation must be improved. Providing insufficient information about what is being shared and with whom (and in some cases, none at all) is not a future-proof option.

We therefore recommend conducting a survey of best current practices for user interface design for authorization handling in OAuth. We believe that the insight gained will also be helpful with the work on other identity management solutions since these types of dialogs are needed as a building block.

### *C. Standardized JavaScript Crypto Library Support*

Many OAuth implementations make use of JavaScript, both in the browser and on the Web server side. The development of OAuth requires that certain cryptographic functions be reused. Instead of re-implementing this functionality in JavaScript, at the expense of both security and performance, we suggest standardizing crypto API support in JavaScript.

A starting point for standardization activities could be [8].

### *D. Moving Crypto Into the Browser*

OAuth requires clients (third parties) to use authenticated requests to protected resources. These requests use different forms of authentication schemes for client/servers taking part in OAuth, such as bearer tokens [9] and MAC authentication [10]. While the cryptographic functionality can be implemented in libraries, we believe that cryptographic functionality should be moved into the browser implementation. Without baseline support in the browser itself, Web application developers are tempted to implement their own crypto-support in their applications. Historically, these attempts have led to weakened security. Implementing cryptographic algorithms is complex; cryptographic routines must come from established, well tested sources, and not be written as needed.

[11] provides valuable input on this topic.

## III. Conclusion

In this position paper we focus on a single Web identity management protocol. OAuth is a widely accepted solution for securing cross-domain resource sharing on the Web. Improving the support in the browser will help to increase security of the Web identity management eco-system.


## IV. Acknowledgments

We would like to thank Axel Nennker, Anders Rundgren, Paul Madsen, Igor Faynberg, Eran Hammer-Lahav, Torsten Lodderstedt, David Dahl, and Dave Nelson for their review comments.

## REFERENCES


[1] E. Hammer-Lahav, D. Recordon, and D. Hardt, "The OAuth 2.0 Authorization Protocol," Apr. 2011, IETF draft (work in progress), draft-ietf-oauth-v2-15.txt.

[2] IETF, "Open Authentication Protocol (oauth) Working Group Charter," Apr. 2011, http://datatracker.ietf.org/wg/oauth/charter/.

[3] "Addressing the "Web Security Standards Deficit"," Apr. 2011, http://webpki.org.

[4] P. Hoyer, M. Pei, and S. Machani, "Portable Symmetric Key Container (PSKC)," Oct. 2010, RFC 6030, Request For Comments.

[5] IETF, "Application Bridging for Federated Access Beyond web (abfab) Charter," Apr. 2011, http://datatracker.ietf.org/wg/abfab/charter/.

[6] J. Howlett, S. Hartman, H. Tschofenig, and E. Lear, "Application Bridging for Federated Access Beyond Web (ABFAB) Architecture," Mar. 2011, IETF draft (work in progress), draft-lear-abfab-arch-02.txt.

[7] B. Leiba, "OAuth Additional Security Considerations," Mar. 2011, IETF draft (work in progress), draft-leiba-oauth-additionalsecurityconsiderations-00.txt.

[8] "JavaScript crypto," Sep. 2010, http://tinyurl.com/dx4x5w.

[9] M. Jones, D. Hardt, and D. Recordon, "The OAuth 2.0 Protocol: Bearer Tokens," Mar. 2011, IETF draft (work in progress), draft-ietf-oauth-v2-bearer-04.txt.

[10] E. Hammer-Lahav, "HTTP Authentication: MAC Authentication," Jan. 2011, IETF draft (work in progress), draft-hammer-oauth-v2-mac-token-02.txt.

[11] IETF, "DOMCrypto API," Apr. 2011, https://wiki.mozilla.org/Privacy/Features/DOMCryptAPI#Designs.