OAuth is a specification that allows “secure authorization in a simple and standard method from web, mobile and desktop applications.” It was built specifically to work with HTTP and allows a third-party site to obtain information about a user from a second site—for instance, Google or Facebook—without requiring the user to provide access credentials to the third party. Essentially, it allows users to approve an “application to act on their behalf without sharing their password” in order to share data (e.g., posts, photos, etc.) from one site to another.

An open standard, OAuth 2.0 is geared toward consumer developers who want to interact with users’ protected data or service-provider developers who support users and who want to share data from the service in web, mobile, and other apps and widgets. Technically speaking, with OAuth, the third-party application redirects the user’s browser to the identity provider (e.g., Twitter)—what OAuth calls the “Authorization Server”—with parameters in the URL that indicate what access is being requested. An authorization code is then sent back to the app; the code can then be exchanged for a token (which has a limited lifespan) that grants the app access to the user’s data at the provider site.

OAuth is just one approach to authorization in the federated identity space. SAML (Security Assertion Markup Language) is a standard predating OAuth with significant deployment among enterprises and education. OAuth, like SAML, requires additional profiling to be interoperable, and OpenID Connect and UMA are profiles of OAuth with significant mindshare.

Importance to Higher Education

Many higher education vendors—including Sakai, Salesforce.com, Dropbox, SharePoint, and Google Apps for Education—make use of OAuth; if you have any of these on your campus, you already have systems that use OAuth. OAuth bypasses university data stewards (such as the registrar, HR, office of research, etc.) by enabling individual end users to release credentials to third parties. By putting the decision for information release in the hands of users, however, the campus may lose some control over what password policies are in place. As a result, for higher-risk applications this approach might not provide sufficient security unless the institution has control over how and when passwords are managed.
InCommon, the U.S. higher education federation, has recently begun work to pilot a social gateway service that allows OAuth to be integrated into the broader set of InCommon SAML services now provided to the community. In the long term, the social-to-SAML gateway allows institutions to use OAuth to extend their campus authentication infrastructure; in the short term, the real relevance and importance of these approaches means that campuses can take a simple step (obtain the use of the gateway) to interoperate with nontraditional communities (such as applicants, parents, alumni, and others) so that they can access campus services without requiring the campus to issue identities to them.

Because OAuth serves as another tool in the identity toolbox, it is important for higher education institutions—particularly those working to develop apps for their users and supporting app development by their researchers and students—to be aware of OAuth’s capabilities and risks.

**Current Landscape**

Increasingly, institutions are faced with the challenge of extending services to new constituents such as alumni, prospective students, and lifelong learners. OAuth, in combination with InCommon and social gateways, allows campuses to provide a holistic solution to their identity management needs. Another important driver in the adoption of OAuth is its large installed base of developers. However, OAuth is currently in place or in development at fewer than 12% of the institutions surveyed, with an additional 14% tracking this technology (see figure 1).

![Figure 1. Results of the 2015 Top 10 Strategic Technologies survey, when asked about OAuth](image)

Work on OAuth began in 2006, and version 2.0 was released in October 2012. However, security flaws exist; notably, the “Covert Redirect” security flaw was exposed in spring 2014, which works by intercepting requests from an OAuth client and redirects the user to a malicious location. In addition, although the OAuth 2.0 specification describes a number of pieces that make up a protocol—such as architectural components, message formats, and high-level integration patterns—it doesn’t explicitly state the correct way for these pieces to come together and interact with each other. This gap means developers who implement OAuth 2.0 can interpret the specs differently and may create incompatible (and possibly incorrect) versions of OAuth solutions.
When You Should Expect It

The main question for institutions is how the campus IDMS can use OAuth, InCommon, and other tools to support their institutional identity needs. There may need to be additional monitoring infrastructure in place before widespread OAuth adoption can take place. Campuses will need to consider what the lack of a trust fabric means and whether they want to allow their students to access (via OAuth) any application on the web.

However, as identity provider operators, some institutions may feel the need to support OAuth sooner than later to facilitate SSO-integrated mobile application development/ adoption and to integrate with cloud services. As such, campuses may want to consider beginning pilot support for OAuth, but it is recommended that its adoption be limited to low-risk services at this time.

Reviewers

ECAR would like to thank the following people who provided their very helpful and significant feedback on this paper: Scott Cantor, Senior Systems Developer, The Ohio State University; Steven T. Carmody, IT Architect, Brown University; John (Jack) Suess, Vice President of IT and CIO, University of Maryland, Baltimore County; and Albert Wu, Senior Director, Information Management Services, UCLA.

Notes

1. See http://oauth.net/.
2. See https://dev.twitter.com/oauth/overview/faq.
4. Because access tokens expire (the longevity is configured in the expires_in property of the standard), refresh tokens are also issued. These allow the third-party app to request a new access token when the initial token has expired. Token expiration is a security measure against fraudulent access. For more information, see http://stackoverflow.com/questions/3487991/why-does-oauth-v2-have-both-access-and-refresh-tokens.
5. See https://developers.google.com/accounts/docs/OAuth2WebServer.
7. See http://openid.net/.
8. UMA (User-Managed Access) is a profile of OAuth; see http://kantarainitiative.org/confluence/display/uma/Home.
9. With a personal Google account, for instance, you can manage which third parties you release information to, but with Google Apps for Education, the institution sets administrative controls over which third parties are able to interoperate with the campus GAE domain. The institution can choose “all” (which may introduce risk) or individually enable specific apps (which is not a scalable process).
10. OAuth has a large developer base in part because “using OAuth” means that it has to be embedded into a program, which means that the ability to move to other paradigms or protocols is lost. Some SAML implementations (e.g., Shibboleth) exist completely outside the application; the application isn’t modified to leverage it.