

■ KENNETH E. KENDALL, Feature Editor, School of Business-Camden, Rutgers University

More often than not, we find that the intersection of human behavior and software is worthwhile in the way it facilitates communication, improves decisions, and bolsters infrastructure. Unfortunately, Web-based human-computer interaction also provides ample room for confusion, errors, and even fraud. In this month's exciting column, our contributor Tim DiVito explores an all-too familiar problem of the growing proliferation of passwords needed to conduct secure ecommerce transactions on Web sites. In this instance, both humans and software bear responsibility where ecommerce Web site security is at stake. Our contributor examines current solutions for authorization and authentication, and then investigates a new technology called OpenID that shows great potential to solve some of the authentication problems, which only show signs of increasing over time. While OpenID exhibits promise, Tim offers valuable insights into the barriers to implementation that lie in wait for the unsuspecting adopter. [Kenneth E. Kendall, Feature Editor]

OpenID: A Potential Authentication Technology

by Timothy DiVito, Rutgers University

How many username and password combinations do you use on the Web, 10, 25, 100 or more? OpenID is an emerging technology that provides an easy way to take all of these passwords and combine them into a single set of credentials that you can use on any OpenID-enabled Web site. OpenID is not a specific kind of authentication, but a framework that allows Web developers to program their sites to pass on the job of authentication to an external trusted OpenID provider who then passes back the authenticated user for authorization.

This technology, if implemented, should make it easier for users to keep all of their credentials safer and more secure. Because OpenID lets users choose which provider they wish to use, they are allowed to single out a provider that is trustworthy, thereby making their information safer. Furthermore, since users only need to give a secure password to a single site, their password is kept more secure than if they had given it to many different sites. This should ultimately lead to a more seamless user experience for all users of OpenID.



Timothy DiVito

is the web developer for Rutgers University School of Law-Camden. He holds a BA in information technology and informatics from the Rutgers School of Communication, Information

and Library Studies and is less than a year away from finishing his MBA from the School of Business at Rutgers-Camden. Mr. DiVito was the first managing director of the Livingston Theatre Company in Piscataway, NJ, where he was an executive producer of 12 full-scale 'Broadway Style' musicals. Mr. DiVito also owns a small website design and consulting company dedicated to giving small businesses low-cost eCommerce and IT solutions. Tim also volunteers his time as a firefighter for the Haddon Heights Fire Department in Haddon Heights, NJ.

tdivito@camlaw.rutgers.edu

OpenID is a technology developed by Brad Fitzpatrick, formally of Live Journal, to simplify the user experience by allowing one single set of online credentials to control many different accounts online [9]. This technology may not yet be ready for high-value accounts like banking institutions and credit card companies and does not currently have many active sites; however, this emerging technology is definitely worth using to create accounts that are currently available.

How Web Site Authentication Works

Before explaining how OpenID works, it is important to understand how the process of Web site authentication works. There are two separate processes that occur when a user attempts to sign in using a username and password on a basic Web site. First, is the process of Authentication, which is a "process of attempting to verify the digital identity of the sender of a communication such as a request to log in" [6].

This means that when users ask to see information on a Web site that is not available to everyone, they are usually asked to create a username and password for the account known only to that individual.

On subsequent visits, he is asked to re-enter that username and password. The process of authentication occurs after the user enters both pieces of information and clicks 'LOGIN' on the page. Those two strings of text are sent to the site and compared to the original username and password created on the account. If they match, the site assumes that the person typing at the computer is the same person that created the account, and the site then proceeds to the Authorization process. If the information does not match, the user is forwarded back to the login page and asked to try again.

Authorization is a process "that protects computer resources by only allowing those resources to be used by resource consumers that have been granted authority to use them. Resources include individual files, data, computer programs, computer devices and functionality provided by computer applications" [7]. So, even after the system figures out that the person typing on the computer is the correct user logging in, it still needs to decide what information it is able to give that person.

For example, a person signed-in to a social networking site like Facebook may be authorized to see her own personal contact information but would not be allowed to gain access to the personal information of any of the other users. However, the administrators of the software would be authorized to access this information if they needed it. Some sites implement the process of authorization very loosely, and some have fine-grained permissions set for each account to ensure data isn't passed from one user to another without permission, but both processes are needed before a username and password can be accepted by a Web site.

How OpenID Authentication Works

OpenID is solely designed to handle the Authentication part of the process [5]. When users enter their OpenID URL, they are instantly forwarded to their personal OpenID provider, who then authenticates them using whatever method they choose. Once the site is satisfied that the user is who he is asking to sign-in as, he is forwarded back to the original site, along with any information the user wants to send to the site, like email, phone number, address, and so on.

Andrew Conray-Murray [1] from *InformationWeek* gave a very concise example on the basic technical process in which OpenID operates:

OpenID 2.0 has three basic elements: a user with a Web browser (User Agent); a Relying Party (the Web site the user wants to log on to); and an OpenID Provider, which asserts that the user owns a particular URL. The OpenID Provider may also possess a variety of identity elements, such as a user's name, date of birth, and e-mail address (see diagram, p. 64). When a User Agent signs on to a Web site with an Identifier (a URL), the Relying Party contacts the Provider for an assertion that the user owns the Identifier. Messages are exchanged using HTTP Post and Get. OpenID uses Diffie-Hellman key exchange to negotiate a shared secret to sign communications.

When a Relying Party contacts the OpenID Provider, the OpenID Provider asks the user to authenticate, then confirms which identity information it should send to the Relying Party. If the user consents to provide the identity elements requested by the Relying Party, the OpenID Provider sends them. The Relying Party processes the elements, and the user is logged in. [1, p. 64].

OpenID version 2.0 is currently in the process of being implemented. It has a more detailed set of directives that can be called by the relaying party for more specific information.

Benefits of OpenID Authentication

OpenID is beneficial to the end users in the following ways: First, users are able

to use a single set of credentials for the majority of their Web accounts, eliminating a great deal of confusion, as well as the time needed to try multiple username/password combinations until the correct one is found.

Second, users are able to choose their own OpenID provider. The framework is open source, so there are an unlimited number of possible providers. Users are even able to *set up their own OpenID system, given the proper technical skills*. This means that the user is able to place their trust in a specific provider, maybe one with a long-standing history, a company like VeriSign or AOL, instead of some random, possibly unreliable site.

Third, the security of the user's password itself is enhanced. Since most people do not use a new, separate password for each and every online account they have, it would be inherently more secure to expose your password to one trusted provider instead of many different Web sites, each with varying levels of security.

Potential Barriers to Implementation

The OpenID framework has many potential downsides. They include the already pervasive use of username/password combinations already in use as well as the vulnerability of the system to Phishing attacks.

According to an article in *The Times of India*, before removing the exact number of pages that it indexed, Google boasted over 8.16 billion Web pages in its index [4]. In the highly unlikely event that Google stopped indexing any new pages on that day, and only one hundredth of those pages required login credentials, each having no more than 250 user accounts, there would be over 20.2 trillion different user accounts in the world today.

The general public has become complacent with the current process of authentication, and it is going to be difficult to change minds, as well as perceptions, on how the technology should function. In addition, even after the average Web surfer is convinced of the

benefits of the system, they still have to wait for the Web sites themselves to spend the time and money to convert their system to using OpenID, which leads to problems in and of itself.

If potential customers want to create an account on a Web site, and they do not yet have an OpenID, is the site going to start forcing people to obtain an OpenID before using the system? Will that drive people away? Even if the end user is content with creating an OpenID, is the site going to recommend a specific provider, or just let people try to find one on their own? What happens if the site recommends a provider that proves to be untrustworthy? Is the site then liable for damages to all of the end users affected should something go wrong? Clearly, OpenID needs to be implemented slowly, so that users are encouraged to embrace the technology and not forced to use it, even at their own peril.

Ever since computer viruses first came to light around 1982 [5], hardware and software designers have been plagued by malicious programs designed specifically for wreaking havoc on computers. In 1990, a new kind of cyber attack, called Phishing, entered the arena. Its first instance was tracked to America Online, where AOL users tried to get other users to give them their AOL passwords by using tricks to make the user think that they were actually talking to a real AOL representative, when in fact their passwords were being compromised [2, 3]. Recently, Phishing has become a serious problem.

Spammers are sending messages to millions of accounts, hoping to get account passwords and credit card information. It has gotten so bad that now programs can be purchased with the sole purpose of looking for Phishing attacks. OpenID is vulnerable to these attacks as well. The difference, however, can be drastic if the user is using the technology heavily. It would be bad enough to find out that a single user account has been hacked, but unthinkable to discover that your OpenID password has been stolen, opening your

account on many different Web sites. So, the barrier to implementation of OpenID in this case is the risk that a single stolen password can be used to compromise many different accounts all at once. Of course, this risk is minimal if most users use only a single password to sign in to all of their accounts anyway.

Alternatives to OpenID

There are a few different alternatives to the OpenID process that also try to enhance the user experience of authentication. Automatic form fillers are the first example. These are programs that are installed on a single computer that allow the user to use a single master password to manage all of their accounts on that computer. Roboform (www.roboform.com) is one of the most popular form fillers. It was originally designed to be installed on a single computer, but has now expanded to a portable application that can be placed on a USB drive and taken to multiple computers. Password Safe is another program similar to Roboform but is available for both Windows and Linux operating systems. These programs do not suffer from Phishing attacks, but can be hacked or mutated by viruses and spy-ware and have their own security concerns.

In addition to third-party programs for password security, most modern Web browsers will save passwords previously entered into their browser, some even have this feature turned on by default. The biggest problem with this feature is the inability to use the passwords on other computers. OpenID can be used from any computer at any time, while browser-based passwords can only be used on a single computer, and are not easily transferable. In addition, users would be locked into using that browser if it has the saved password information and would be unable to change to another browser without re-entering the information.

OpenID is able to be used from any browser that will support Web-based forms. This not only includes Windows, Mac and Linux, but also mobile phones

and portable handhelds that are unable to have software installed on them. Microsoft has also committed themselves to this technology by adding it to their newest version of Windows, Vista, with a program called Card Space. OpenID even works on the iPhone!

Recommendations

Since OpenID is a free service, available to anyone, there is no reason not to get an OpenID now. How many times have you tried to sign up for an account on a Web site, just to get your potential username rejected because someone already has it? It is important to do research on some of the main providers of OpenID. If, for example, you have an account with AOL or AOL Instant Messenger, you already have an OpenID which is available at

<http://openid.aol.com/screenname>

This is also the case with Live Journal, Technorati, Vox, and WordPress, which are all OpenID providers. If you want a new account, here is a list of well known providers from the OpenID Web site.

- VeriSign Labs (<https://pip.verisignlabs.com/>)
- ClaimID (<https://claimid.com>)
- Myid.net (<https://myid.net>)
- MyVisopp(<https://myVidoop.com>)

Go to openid.net/get to get a full list of providers.

One of the advantages to choosing VeriSign Labs as your OpenID provider is their commitment to an enhanced process called two-factor authentication. Two-factor authentication uses more than one piece of information to authenticate you. For example, I was able to purchase, for a small fee, the small key ring dangle shown in the photo below.



When the button on the dongle is pressed, a number is generated that VeriSign is expecting. After 30 seconds, that number is not valid and a new one needs to be issued to login. This makes it almost impossible for a hacker who is remotely stationed in China to gain access, even if they were able to get both your password and your secret code.

This type of authentication combines 'something you know,' which is your password, with 'something you have,' being the dongle itself. VeriSign also offers a way for users to register their cell phone to receive a text message with this type of code in the event that they lose their dongle and need access to their account. Other OpenID providers may be doing similar things, but so far, it seems that VeriSign is one of the most secure providers available.

Conclusions

The OpenID framework has been designed to give end users an easier way to manage multiple accounts online while also giving them a choice as to whom they will trust with their most sensitive information. This emerging technology should make signing into minor accounts much easier, but it is questionable whether it is a technology that can be implemented for sites containing sensitive information such as credit card statements and banking information. The potential for that kind of pervasiveness is there, but it would require an enormous amount of trust from the end user to actually make this happen.

References

[1] Conray-Murray, Andrew. (2007). "Single sign-on for the Web?" *Informationweek*, Sept. 3, 64, 68.

[2] Krebs, Brian. (2004). A brief history of phishing. *The Washington Post*, Nov. 18. Last accessed on 30 Oct, 2007.

<http://www.washingtonpost.com/wp-dyn/articles/A59350-2004Nov18.html>

[3] Markoff, John. (2005). "Prized query: How many pages does Google have?" *The Times of India*, Sept. 28. Last accessed on 28 Sept. 2005.

<http://timesofindia.indiatimes.com/articleshow/1244851.cms>

[4] Ramzan, Zulfikar. (2007). A brief history of phishing, Part 1. *Symantec*, August 10. Last accessed on 30 Oct, 2007.

http://www.symantec.com/enterprise/security_response/Weblog/2007/08/a_brief_history_of_phishing.html

[5] Swartz, Jon. (2007). Technology cuts down on Web registrations. *USA Today*, March 16, 1b.

[6] Wikipedia, The Free Encyclopedia. (2007). Authentication. Wikimedia Foundation, Inc.

Last accessed on 25 Oct 2007.
<http://en.wikipedia.org/w/index.php?title=Authentication&oldid=166988966>

[7] Wikipedia, The Free Encyclopedia. (2007). Authorization. Wikimedia Foundation, Inc. Last accessed on 31 Oct 2007.

<http://en.wikipedia.org/w/index.php?title=Authorization&oldid=166522494>

[8] Wikipedia, The Free Encyclopedia. (2007). Elk cloner. Wikimedia Foundation, Inc. Last accessed on 31 Oct 2007.

http://en.wikipedia.org/w/index.php?title=Elk_Cloner&oldid=160295304

[9] Wikipedia, The Free Encyclopedia. (2007). OpenID. Wikimedia Foundation, Inc. Last accessed on 31 Oct 2007.

<http://en.wikipedia.org/w/index.php?title=OpenID&oldid=167809765> ■

Submitting articles to *Decision Line*

Members are invited to submit essays of about 2,000 to 2,500 words in length on topics of their interest, especially articles of concern to a broad, global audience. Please send essays (including brief bio and photo) to either the respective feature editor or to Editor Krishna Dhir.

Deans' Perspective & Editor

Krishna S. Dhir, Berry College
kdhir@berry.edu

Doctoral Student Affairs

Xenophon Koufteros, Texas A&M University
xkoufteros@mays.tamu.edu

E-Commerce

Kenneth Kendall, Rutgers, The State University of New Jersey
ken@thekendalls.org

From the Bookshelf

Peter Ittig, University of Massachusetts, Boston
Peter.Ittig@umb.edu

In the Classroom

Bih-Ru Lea, Missouri University of Science and Technology
leabi@umr.edu

Information Technology Issues

Vijayan Sugumaran, Oakland University
sugumara@oakland.edu

In the News

Carol Latta, Decision Sciences Institute
clatta@gsu.edu

International Issues

John Davies, Victoria University in Wellington, New Zealand
john.davies@vuw.ac.nz

Membership Roundtable

Gary Hackbarth, Northern Kentucky University
gary.hackbarth@nku.edu

Production/Operations Management

Daniel A. Samson, University of Melbourne, Australia
d.samson@unimelb.edu.au

Research Issues

Miles Nicholls, RMIT University, Australia
miles.nicholls@rmit.edu.au