Browser Assisted Identity Management

Yu Wang and Aanchal Gupta
Yahoo! Inc
701 First Ave
Sunnyvale, CA 94089
yuwang@yahoo-inc.com, aanchal@yahoo-inc.com

***Abstract*— This paper describes significant advantages in providing browser-assisted identity management and outlines what can be done by browser/browser plugin vendors, identity providers and relying parties to make this happen.**

Keywords-identity management; identity provider; relying party; OpenID; OAuth; security, privacy; WebFinger

I. INTRODUCTION

In the recent years there has been a plethora of web sites that provide personalized services to Internet users. User authentication has become part and parcel for building these web sites. Because of the complexity of managing user authentication directly, some web sites have chosen to become relying parties by accepting users from independent identity providers (IDPs) such as Yahoo!, Google, Facebook, and AOL through either open authentication protocols like OpenID or proprietary ones like Facebook Connect. In addition to user identity and basic user data, many IDPs also have a lot more information about their users' activities, friends, etc and allow these relying parties to tap into this data. Relying Parties can use protocols such as OAuth or Facebook Connect to access this data and at the same time increase the value of the identities managed by these IDPs. This has helped in reducing the number of entities a user has to manage. However we still see quite a few difficult issues not well addressed to date.

First, many sites have implemented their own identity provider selection mechanism and different implementations have caused inconsistent user experience. For example, even for sites that accept OpenID, the sign-in UIs are quite different and users have to get reoriented whenever they visit any such web sites. Following are some examples:

http://www.huffingtonpost.com/users/login/

http://www.winamp.com/user/snslogin

https://www.plaxo.com/auth?src=header&secure=1

Second, many phishing web sites have been set up to mimic legitimate IDP sites to steal user credentials. Even though user education is critical here, past experiences have told us that it is indeed hard for users not to fall into these traps. To complicate the issue further, some web sites ask users for their IDP account name and password directly to get access to users' data, either because of no suitable APIs from IDPs or sloppy security practice. Users are conditioned to think sometimes they have to disclose their IDP account name and password to third-party web sites directly which is a sad reality.

We believe that Web browser can mitigate the security and UI issues with the support from both IDPs and RPs. The rest of the paper is organized as follows: In Section II, we explain why browser can be an integral part of identity management. Sections III and IV describe what IDPs and RPs can do to facilitate browser-assisted identity management respectively. Section V concludes this paper.

II. BROWSER SUPPORT FOR IDENTITY MANAGEMENT

First, it is important to understand that Web browser is much more trustworthy than any web site because it intermediates between users and web sites and has easy access to users' all online credentials. Storing credentials in browsers is usually a much safer choice than storing them online through web sites that provide such service. In the former case, users only need to worry about their Web browser's security. In the latter case, users also have to worry about whether those web sites that manage their credentials are secure or not and whether those web sites have applied the latest security fix timely.

Second, Web browsers usually have automatic update feature to allow users to get the latest security fix conveniently. Once user turns it on, users don't have to worry much about if their browser is up to late with the latest security fix. In contrast, it may be much harder for some web sites to do any security upgrade quickly for fear of breaking some existing functionality or because their systems are too old to upgrade. Especially in some shared hosting environments, web site owners are stuck with the software packages provided by their hosting service providers, e.g., they may be stuck with PHP 4.

Third, Web browsers can store and access users' credentials, preferences and etc easily, i.e., they know a lot more about their users and hence can guide user with smooth and secure Web experience. Suffice to say we can depend on browsers' trustworthiness to assist user with identity management in the following ways. Later we will describe what IDPs and RPs can do to support this joint effort.

A. Implement a native IDP selection interface

RP sites can advertise the IDPs they support through well-known locations and when a user first visits these web sites, a Web browser can show a native IDP selection interface to ask the user to choose which identity the user wants to use with these sites. This can be implemented either natively in the browser or through browser plugins.

Web browser/browser plugin should probably give those IDPs who have established a certain level of trustworthiness preferential treatment, because major IDPs probably don't want to see themselves listed together with mr-nobody.com sites.

Such trustworthiness can usually be established by implementing industry-endorsed security and privacy features, e.g.,

- User credential submission is through HTTPS.
- Provide reasonable measures for anti-phishing (frame busting and etc).
- Any sensitive operations should require user confirmation.

B. *Enforce IDP security policies*

Web browser/browser plugin can store a user's identities and credentials (password, tokens) as long as they have the user's approval and can manage them securely. However they need to provide reasonable assurance that they don't submit stored credentials to phishing sites or other unintended sites. One way is for Web browsers to subscribe to reputable services that warn about a phishing site that a user intends to visit. Another way is for IDP to advertise their security policies at well-known place that Web browsers should enforce unless explicitly overridden by users. For example, an IDP can advertise that

- Its users' credentials should never be submitted from a window that is iframed in a window belonging to a different domain
- Its users' credentials should always be submitted through HTTPS and to specific sites, e.g., https://login.example.com/
- It discourages browsers from storing users' passwords. Instead it may suggest browsers store some kind of tokens that are obtained first through username/password validation and then be stored. Later the token can be used to obtain session credentials to sign a user into IDP web site. The advantage of using tokens is that users can grant different tokens for different web sites and can revoke tokens for a web site without changing their password or impacting other web sites.

C. *Manage user's identity preferences through browser*

Web browser can learn about users' identity providers and RP web sites by tracking from which sites users obtain sign-in credentials and which sites accept users' credentials. This should be subject to users' approval though. Then such information can be stored in browser's offline store. Brower vendors can agree upon a common access API and open it to all browser plugin developers even if browser vendors don't necessarily want to manage user's identity natively.

In addition, it is desirable that browsers should allow users to export their identities and preferences. This allows users to migrate/access their identities and preferences in other environments.

III. IDENTITY PROVIDER SUPPORT

We have described what browsers can do to assist user identity management. Now we are ready to describe what IDPs can do to facilitate identity management.

First, IDPs can advertise what authentication protocols they support. Browsers can then match them against RP sites and show it in the identity selection interface generated by the browser. Suppose a browser knows that its user has signed into Yahoo! before and Yahoo! advertises that it supports OpenID as an IDP. When the user visits a web site that supports OpenID as an RP, then the browser can automatically add the user's Yahoo! account as one of the identities to be chosen to sign into this web site.

Second, IDP can advertise what authorization protocols they support. This applies to those IDP sites that make a distinction between authentication and authorization. For example, the OAuth protocol is often used for authorization of access to protected user data. Browser can mandate that only OAuth protocol is used for authorization for this IDP and no website should collect IDP's account name and password. Users can be alerted with a warning in browser chrome if any website tries to do this for such IDPs.

Third, IDP can publish their APIs, security and data access policies and etc at well-known locations through open protocols such as WebFinger. As it has been stated earlier, IDPs may indicate that

- It doesn't allow iframed sign-in
- Its user credentials should be submitted through HTTPS only
- Re-authentication or user confirmation are needed for certain operations

Surely browser should still allow users to override these policies but should give proper warnings beforehand.

Fourth, IDP can also advertise the URL where a new account can be created. This is useful when certain RPs advertise that they support these IDPs that the user doesn't have any account yet.

Fifth, IDP can also provide service to inform the browser whether a user maintains a valid sign-in session or not. This can help to achieve one-click sign-in if a user chooses to do so.

IV. RELYING PARTY SUPPORT

We have described what browsers and IDPs can do to assist user identity management. Now we can describe what RPs can do to support this joint effort.

First, RPs can state which IDPs they accept through well-known locations so browser can present a unified sign-in interface to users. Though RPs still have to build their sign-in interface separately for those browsers which don't support identity management, at least RPs don't have to spend the same amount of the effort to make the UI acceptable to all browsers.

Second, RPs can advertise the profiles that they prefer IDPs to support and browsers can put those conforming IDPs more prominently at the identity selection interface.

Currently there is an ongoing effort to get IDPs certified for different levels of profiles:

https://sites.google.com/site/oauthgoog/Home/certificationchecklist

By requesting IDPs that provide higher level of profiles RPs may expect that their users can enjoy better and more secure sign-in experience and they can get user data easily. For example, some profile requires the IDP to have high availability, strong authentication scheme and friendly pages for mobile and non-Javascript browsers.

Third, RP can advertise the endpoints that solicit and accept IDP credentials. Web browsers can then choose to implement some authentication/authorization protocols natively and fast track users' sign-in experience.

V. CONCLUSION

In this paper we have explained how browser-assisted identity management can provide better user experiences and more secure authentication and authorization services for end users and described in detail what browsers, identity providers and relying parties can do together to make this happen.