Dear OpenID committee, Google, and PayPal Developers,

We are a research team doing security research. Recently, we studied the OpenID services of two prominent providers: Google and PayPal. We found a serious flaw which allows unauthorized party to log into victim user's account in the relying party websites.

An Example OpenID Request

https://www.google.com/accounts/o8/ud?openid.ns&...&openid.ext1.type.email=http%3A%2F%2Fschema.openid.net%2Fcontact%2Femail&openid.ext1.type.firstname=http%3A%2F%2Faxschema.org%2FnamePerson%2Ffirst&openid.ext1.type.lastname=http%3A%2F%2Faxschema.org%2FnamePerson%2Flast&openid.ext1.required=email%2Cfirstname%2Clastname

An example OpenID Response

rp.com/login?openid.ns&…&openid.ext1.type.firstname=http%3A%2F%2Faxschema.org%2FnamePerson%2Ffirst&openid.ext1.value.firstname=Alice&openid.ext1.type.email=http%3A%2F%2Fschema.openid.net%2Fcontact%2Femail&openid.ext1.value.email=alice.the.hostia%40gmail.com&openid.ext1.type.lastname=http%3A%2F%2Faxschema.org%2FnamePerson%2Flast&openid.ext1.value.lastname=Hostia

We believe that this is a very generic issue. Currently it has been confirmed on shopgecko.com/store/ (a sample website of Magento shopping software) using PayPal Access and toms.com using Google ID. Toms.com is a shopping website that allows several SSO schemes. Our exploit targets its Twitter users.

To describe the problem, let's assume Alice is a victim user, and Bob is the attacker. Before explaining these two cases, let's take a look at a typical OpenID authentication response, which is shown above. In the response, an argument of particular interest is openid.ext1.value.email (value.email in short), which the relying party website (RP) treats as the email address of a user. However, Google, the identity provider (IdP), thinks differently. It actually sets the element's value according to openid.ext1.type.email (type.email in short), an element in the OpenID request. The RP of course sets type.email to be http://schema.openid.net/contact/email, OpenID's type for user emails. However, a malicious user can set it to be anything, such as http://axscheme.org /namePerson/first (OpenID's data type for first names). Therefore value.email in OpenID response can hold the user's first name. An exploit would be possible if he could register with Google his first name as "alice@a.com". Because many OpenID-enabled websites use the registered email of a user as her authentication token. This data type confusion can lead to signing Bob onto Alice's account. We confirmed that smartsheet.com indeed takes Bob's first name as an email under the exploit. We

believe that the misunderstanding about the content of value.email is pervasive, given that Google developer's guide only uses value.email as an example of requested user attributes in its specification, and never mentions how its content is actually determined [1].

However, this exploit did not get through completely, because Google ID's user registration page does not treat "alice@a.com" as a valid first name. Therefore, a natural question produced by our analysis is whether there is a way to register "alice@a.com" as the value of any non-email field in Bob's Google ID profile, maybe through direct API calls instead of the user registration page.

We now can explain the two websites which we have end-to-end exploits. shopgecko.com/store/ identifies a user by his/her PayPal ID, which is not a secret. The type of the ID is https://www.paypal.com/webapps/auth/schema/payerID, but Bob sets it to http://schema.openid.net/contact/street2, which is the type of the second line of a user's mailing address. We successfully registered a user whose second line of mailing address is identical to Alice's PayPal ID. For toms.com, we found the element called "email" in fact contains a user's Twitter ID when it is doing Twitter SSO, although it contains the user's email address in other schemes, such as Google ID. Bob, a Google user, can register his first name as "AliceOnTwitter", which is Alice's Twitter ID, and sign in as Alice through Google.

We have made two video demos showing the attacks.

Toms.com attack demo: http://www.youtube.com/watch?v=5MX5ChOX5ic

Shopgecko.com demo: http://www.youtube.com/watch?v=cp9SLJO1rI4

This email is cc'd to Google, PayPal, toms.com, and Magento. Please feel free to let us know if you need more details. We would be appreciated if you can give us a short reply after the receipt of this message.

[1] Google Code. "Federated Login for Google Account Users," http://code.google.com/apis/accounts/docs/OpenID.html

Kind Regards,
Rui Wang, Shuo Chen, XiaoFeng Wang