Notes-ConferenceCall-July-01-2013

Social Identity Conference Call, 07/01/2013

Overview

- Topic: Minimal Gateway Attribute Bundle
- Reference: https://docs.google.com/a/brown.edu/document/d/1VHuga_hifyjQJbr9i17Vw4FEprwLJxM13Kuc-PX_3Zg/edit

Summary

The discussion focused mainly around EPPN and email. The question of whether or not it is acceptable to use email as EPPN was discussed, and the general feeling was this is not a good idea, but it really is up to the SP/application admin to determine how he/she wants to consume attributes. However, some requirements (eg "advance predictability") may make the email choice more attractive. Scott also noted that shibd is highly configurable, so if the Gateway issued an attribute in one format, shibd could easily rewrite it so the SP could consume it in a format it expects.

Lucas noted that the Gateway itself might allow the SP admin to determine what attribute from the social provider should be set as EPPN, and perhaps even the format of the attribute. He offered three suggestions:

1. EPPN as email
2. EPPN as a function of email, with the scope being the social provider, not the gateway
    - example: `lr+lucasrockwell.com@google.com`
3. EPPN as some other unique ID from the social provider, scoped to the domain of the provider

Of course, all of these options may not be available for every social provider, as some do not provide a persistent identifier, some do not provide an email, some provide email but the user can change it, and finally, some offer more than one email. To the last point, Jim noted that Windows Live can return more than one email, and all agreed that if this happens, the Gateway should return all values.

> **Recommendation**
>
> EPPN should be a username, not an opaque ID, and it should not be the user's email address if that can be avoided. In the end, it will probably be up to the SP admin to make the choice, and it is up to us (or the Gateway documentation) to make sure the SP admins understand the ramifications of their choices.

The discussion also focused on the discoverability of a user's account name/persistent identifier. In the case of Twitter, every user knows their username, because that is simply how the service works. Other providers, like Facebook, also have a human-readable username, but in some cases it may take some discovery for a user to figure it out. Facebook also allows a user to change their username (apparently only once). This is not bad, but it is something SP admins should be aware of.

Several options were discussed regarding how to provision a user using their social identifier. Steven gave an example of an application where students will need to grant access to their supervisor from their summer internship, and the supervisor will be logging in via a social provider. Ideally, the user could just enter their supervisor's social username into the application. Of course, this works great for services where you know the username, like Twitter and Facebook. This use case could also be addressed by having the student enter a social email address, and sending an INVITE to that address. The EPPN would be obtained when the user returns.

> **Recommendation**
>
> Instead of asking end users to provide their social identifier (or the identifiers of others), ask them for an email address. Later, when the user logs into your app in response to an email invitation, map the identifier asserted by the social IdP to the email address originally provided by the user.

Also on the topic of attributes, it is possible that instead of EPPN, the new eduPersonUniqueID could be used when the ID is not really EPPN-friendly (like Google's profile ID and the Windows Live ID). Furthermore, these attributes could be used to seed a targeted ID.

Scott mentioned an important item for the group to keep in mind, perhaps for the future: Creating new attributes specifically for social providers. He said, "If we end up passing a Facebook ID around, then we need an attribute for a Facebook ID, and likewise for a Twitter handle and a Google ID." Scott feels this is much better than constructing an EPPN around these values, especially since neither the user nor the social provider will know what you are talking about.

> **Counterexample**
>
> The Google OpenID IdP (and other FICAM-certified social IdPs) asserts a *Private Personal Identifier* such as:
>
> which maps well to EPTID since PPID is per-SP by definition. The problem with asserting PPID as-is is that the target SP is not apparent in the bare string. This is especially important in a gateway environment since the target SP is most definitely **not** the end SP.

Open question: Should the gateway convert PPID to EPTID, and if so, what should the SP entityID be?