Session 4N

IdP Discovery and FedCM

Session Convener: Heather Flanagan
Notes-taker(s): Heather Flanagan

Tags / links to resources / technology discussed, related to this session:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf

Discussion notes, key understandings, outstanding questions, observations, and, if appropriate to this discussion: action items, next steps:

Discussion re: the edu use case with thousands of IdPs and tens of thousands of RPs and how they'll need to be considered with in FedCM's account/session chooser use case.

RP gets a list of IdPs from a third-party federation (in the US, that's InCommon). InCommon manages the metadata (a giant blob of XML from a well-known spot). File of metadata is signed by InCommon and is about 90MB. InCommon serves as the point of trust for IdPs and RPs. You identify an endpoint in SAML metadata that's usually formatted as a URI or URL (includes a domain; see "entityID")

Where does the 90MB file get processed? Used to be the RP would load the entire thing, but that doesn't work any more. It's up the RP middleware to parse the XML server-side and boil it down to a JSON file that would feed their own interface (see the Shibboleth Disco Feed). This hasn't been standardized. Right now, the "nicest" method to do IdP discovery is called SeamlessAccess, and it has nice UI standards for how to find things. It is also protocol agnostic. IdPs have a scope associated with them that's a piece of policy, and InCommon as a cert authority does domain validation to prove control of the domain. If you don't have control of the domain, we aren't going to publish your metadata. The user could type in their email address, and then say chop off everything to the left from the @ sign, but in higher ed that doesn't always work. Colleges and universities may share physical location but have different IdPs, and it's hard for a user to actually guess which email address to use to get access to a specific journal institution tied to the subscription of any one of the schools they are affiliated with. Also, students are more likely to type in their gmail address when asked "type in your email address".

Is there an equivalent to this in OpenID? You can do the exact same thing in OIDC

What parts of this break with third-party cookies? Some discovery components.

There is a proposal to mean it won't 100% break, but the solution will be "crappy" -
Third-party cookies will absolutely go away. That's not optional. If you passed 1000 IdPs to the FedCM API, it would pass 1000 http queries to the IdPs; of those 1000 requests, all would fail except the one you logged into. So, if you're logged into one of the universities, it would come back with your logged in institution. Pretty sure that wouldn't scale. If you send a session identifier to every single server, you'd knock the system over. What FedCM is proposing is an extension to the FedCM is that IdPs can push to browsers to say "I have a logged in user". Where does the API endpoint live? It's a browser API (JavaScript or HTTP header). This is a variation of isLoggedIn. This doesn't help the bootstrap problem of that first IdP discovery.

You can still do the IdP discovery without third-party cookies; it's the persistence found in things like SeamlessAccess that become problematic. FedCM adds a level of efficiency in a different area in that the IdP list, if the user has more than IdP then it would have a curated list of choices per user. Would the new login status be useful independent of FedCM, something that would be lighter weight, something that wouldn't require all IdPs to expose info to the IdP.

University of California system does have an extensive front-channel logout and requires force reauthentication. Action: Heather to get contacts in that system to talk and hopefully help test FedCM

The walled garden of policy is what makes academic federation work. The coherent policy framework makes a huge different.

There are two UX formulations they an offer to capture consents - one that has browser UI in the top right of the screen (URL bar); other is autocomplete.

Some consideration as to how to train people on correct behavior.

Much of what happens next comes from how much the higher ed community can help build and test prototypes.