Implementation Descriptions

Overview

This page lists several of the known OpenID to SAML gateway implementations, and provides information about how they operate.

UW-Madison Development Instance: Social2SAML

Background

UW-Madison chose to go the path of defining separate IdP entities for each of the supported social identity providers for a development instance of a Social2SAML service. The solution was developed with an earlier release of SimpleSAMLphp and the latest version would make the implementation much simpler. We currently have support for Google, Facebook and Twitter. When combined with the Shib SP 2.4+ Embedded Discovery Service, and our own SAML metadata provider, this gives us a way to list the social providers as if they were any other SAML IdP in the selection drop-down box.

SAML attributes returned from each of the three social providers as seen in Apache environment variables:

```
----------
UW-Madison''s Soc2SAML gateway
-----
Twitter:
Shib-Application-ID default
Shib-Session-ID _0ad867d6e583e8f5f897bd102e97f15f
Shib-Identity-Provider https://panda.doit.wisc.edu/idp/twitter
Shib-Authentication-Instant 2012-08-27T15:00:13Z
Shib-Authentication-Method urn:oasis:names:tc:SAML:2.0:ac:classes:Password
Shib-AuthnContext-Class urn:oasis:names:tc:SAML:2.0:ac:classes:Password
Shib-Session-Index _0eeb7aae8714bf1115a36911662025165295a75ae5
Shib-Assertion-Count 00
cn khazelton
displayName khazelton
eppn 16xxxxxx@twitter.com
persistent-id [https://panda.doit.wisc.edu/idp/twitter]! [https://persepolis.wisc.edu/shibboleth-sp]! 8a6bxxxxxxb2a5599564655df607f99e7b220c29

-----
Facebook:
Shib-Application-ID default
Shib-Session-ID _280ef30bdbb6b3d74407bcd98e9a55fc
Shib-Identity-Provider https://panda.doit.wisc.edu/idp/facebook
Shib-Authentication-Instant 2012-08-27T15:04:42Z
Shib-Authentication-Method urn:oasis:names:tc:SAML:2.0:ac:classes:Password
Shib-AuthnContext-Class urn:oasis:names:tc:SAML:2.0:ac:classes:Password
Shib-Session-Index _901fc506235ea6bb89c77a09d7385b9c1065869297
Shib-Assertion-Count 00
cn Keith D Hazelton
displayName Keith D Hazelton
eppn 100002457xxxxxx@facebook.com
givenName Keith
mail khazelton@gmail.com
persistent-id [https://panda.doit.wisc.edu/idp/facebook]! [https://persepolis.wisc.edu/shibboleth-sp]! 725fxxxxxx5afc40fc1af43131c027cac474deda
sn Hazelton
```

```
-----
Google:
Value
Shib-Application-ID default
Shib-Session-ID _d602f68653201efbd494d278c5f02b8d
Shib-Identity-Provider https://panda.doit.wisc.edu/idp/google
Shib-Authentication-Instant 2012-08-27T15:07:23Z
Shib-Authentication-Method urn:oasis:names:tc:SAML:2.0:ac:classes:Password
Shib-AuthnContext-Class urn:oasis:names:tc:SAML:2.0:ac:classes:Password
Shib-Session-Index _6375bb12c3e643580fe4e7b4e710f66cbcd4af4c94
Shib-Assertion-Count 00
cn Keith Hazelton
displayName Keith Hazelton
eppn 58159e6705c4df4eedafce4b56xxxxxx@google.com
givenName Keith
mail khazelton@gmail.com
persistent-id [https://panda.doit.wisc.edu/idp/google]! [https://persepolis.wisc.edu/shibboleth-sp]! dac0xxxxxx2b374401d64924f2b5874b8d691eeb
sn Hazelton
```

IDCorral Shibboleth IdP Proxy

Background

The IDCorral Shibboleth IdP Proxy is a proof-of-concept for proxying OpenID login for Shibboleth-enabled applications. Since it is only a proof-of-concept, it was implemented with JanRain Engage as the mechanism by which a user selects his/her ID provider. JanRain Engage also aggregates OpenID/OAuth /Facebook Connect into a unified API so that the implementing service – in this case the Shibboleth IdP proxy – has a consistant set of attributes to work with.

The original writeup on this concept can be found here: http://lucasrockwell.com/other/idpproxy.html

Attributes Available from Identity Providers

What attributes does the external IDP offer?	The attributes which JanRain Engage offeres up for each provider are listed here: https://rpxnow.com/docs/providers
What attributes do you ask for?	All of the attributes listed under "Basic Profile" are returned regardless.
Which attributes require user consent ?	All of them.
Given Name	givenName (urn:oid:2.5.4.42)
Family Name	sn (urn:oid:2.5.4.4)
Display Name	cn (urn:oid:2.5.4.3) (This should be changed to displayName)
Verified Email	mail (urn:oid:0.9.2342.19200300.100.1.3)
Preferred Username	eduPersonPrincipalName* (urn:oid:1.3.6.1.4.1.5923.1.1.1.6)

*At this time, the eduPersonPrincipalName is set by the user the first time he/she logs in, and the Preferred Username is used as a guide for setting this information. The Preferred Username can not be taken at face value because it is not guaranteed to be unique.

Apache2::AuthAny

Background

The Apache module, "Apache2::AuthAny" was developed as an extensible authentication/authorization system. The module currently protects the Distribute project at https://isds-auth.cirg.washington.edu/distribute/index.php

Apache2::AuthAny creates a set of authentication URLs, one for each provider, that are separate from the location of the protected content. A demo with documentation is available at https://authany.cirg.washington.edu

Attributes Available from Identity Providers

The production version of AuthAny only supports Google authentication through OpenID with the Google email being passed (through AX). The architecture allows any authentication mechanism to be used however, so if another provider is added that passes other attributes, they could potentially be stored in the AuthAny database, and passed through to the protected application.

IdPproxy

Background

IdPproxy was developed as an proxy authentication system, to bridge the divide between SAML2 federations and Social Media. It only works one way, that is it uses Social Media to authenticate persons to SAML2 federation.

Attributes Available from Identity Providers

The focus has all the time been on authentication, gathering identity information has always been more of a 'nice if we get some'. The reason for this is of course the level of assurance that you get for this information. IdPproxy today supports Facebook, Google, Twitter, general OpenID and Windows Live ID. It does not ask for any attributes outside what these services provide by-default. Hence, we have, so far, not tried to find out what is possible to get out of them.

IdPProxy does some transformations of information received from Social Media, like constructing a displayName from given name and family name if no full name was given. It also constructs an eduPersonPrincipalName from an identifier provided by the service and the domain name of the service.

Template

Below is the template to use for listing attributes from the external IdP.

Name of Identify Provider

What attributes does the external IDP offer?	
What attributes do you ask for?	
Which attributes require user consent ?	
(name of input attribute)	repeat this row for each input attribute. This column should contain the name, value, syntax, and semantics of the SAML attribute that you assert using this input value.

Reference links

- OpenID attribute schema: http://www.axschema.org/
- The attributes which JanRain Engage offers for each provider are listed at: https://rpxnow.com/docs/providers

OpenID Simple Registration Extension 1.0

Extract from above:

openid.sreg.nickname	Any UTF-8 string that the End User wants to use as a nickname. [Note: The Profile's Label for this attribute is "Alias/Username."]
openid.sreg.email	The email address of the End User as specified in section 3.4.1 of RFC2822
openid.sreg.fullname	UTF-8 string free text representation of the End User's full name
openid.sreg.dob	The End User's date of birth as YYYY-MM-DD. Any values whose representation uses fewer than the specified number of digits should be zero-padded. The length of this value MUST always be 10. If the End User user does not want to reveal any particular component of this value, it MUST be set to zero. For instance, if a End User wants to specify that his date of birth is in 1980, but not the month or day, the value returned SHALL be "1980-00-00"
openid.sreg.gender	The End User's gender, "M" for male, "F" for female
openid.sreg.postcode	UTF-8 string free text that SHOULD conform to the End User's country's postal system
openid.sreg.country	The End User's country of residence as specified by ISO3166
openid.sreg.language	End User's preferred language as specified by ISO639
openid.sreg.timezone	ASCII string from TimeZone database For example, "Europe/Paris" or "America/Los_Angeles"