What is Federation (and why should I care)?

Rob Crittenden, Red Hat, Inc.
rcritten@redhat.com
Identity

• More than a login id
• You have many:
  – Work
  – Personal interests: sports, hobbies, etc.
  – Software
Federated Identity

- Portability of identity across domains
- Reduces administrative overhead of redundant information
The Parties

- Federation has three participants. I'm going to use SAML2 lingo:
  - Identity Provider (IdP)
  - Service Provider (SP)
  - End user
Identity Provider

• Trusted provider that manages identity information and provides authentication

• Can be part of your infrastructure or a trusted 3rd party
Service Provider

- A Web Server
- Relies on a Identity Provider for authentication
- Can be in or out of your control
Users
Typical Web Usage

User

joe
+ password
juser
+ password
joe.user
+ password
joe
+ password

site1
User DB

site2
User DB

site3
User DB

site4
User DB
The Problem?

• Multiple Passwords
  – Some almost certainly bad
  – Possible re-use
• Remembering which password goes where
• Reliance on each web site protecting its user database
A solution: Federation

● Good for Users:
  - One account* to use everywhere
    • So one password, hopefully a good one
  - Rely on trusted 3rd party to protect passwords

● Good for Web Applications/SPs:
  - Can support additional authentication methods
  - Reduce administrative overhead
    • No user passwords to manage
Federation Highlights

• Trust a 3rd party to do the authentication
• Generally a web-based protocol over TLS
  – Doesn't always require a browser, e.g. rich mobile client
• Centralized or Decentralized
• Common protocols:
  – SAML, OpenID and OpenID Connect
Not Federation

- No control of credentials
SAML 2.0

- Mature
- XML and SOAP over HTTPS
- Centralized: requires agreement between parties
  - Exchange of metadata and public keys
- Single sign on and Single logout
How does SAML login work?

1. Request
2. Redirect to IdP
3. Authenticate
4. Issue Token
5. Redirect to SP
SAML Assertions

• Authentication
  – NameID
• Attributes
• Authorization Decision
• Signed
SAML Single Logout

1. Logout Request
2. Redirect to IdP
3. Find all sessions
4. Logout using SOAP
5. Logout Response for SP2
6. Logout response for SP1
SAML Use Cases

- Typically Enterprise
- Single Sign On
- Employees at Acme, Inc can manage customer relationships at Salesforce using corporate identity
- Book airlines, hotels or cars
SAML Identity Providers

- Shibboleth
- Keycloak
- Ipsilon
- SimpleSAMLPHP
- OpenSSO
SAML Service Providers

- Shibboleth
- Keycloak
- mod_auth_mellon
- SimpleSAMLPHP
- OpenSSO
OpenID

- Decentralized
- Identity is a URL
- You prove that you own that specific URL
- Like SAML, need to trust 3\textsuperscript{rd} party to prove authentication
  - Or, if you want, you can run your own OpenID server
What is an OpenID Identity?

- Contains the location of the Identity Provider
- rcritten.id.fedoraproject.org
How does OpenID work?

1. Login Request
2. Discovery on URL
3. Associate
4. Req. auth
5. Authenticate
6. Validate
7. Issue Token
8. Present Token
OpenID Extensions

• Simple Registration Extension

• Attribute Exchange

• Teams
OpenID Use Cases

• Typically decentralized / user-centric
• Single Sign On
• Use single identity on multiple blogs, forums, e-mail, bug trackers
OpenID Providers

• FedOAuth
• Wordpress-OpenID
• Ipsilon
• MyOpenID (proprietary)
OpenID Relying Parties

- Wordpress-OpenID
- Flask-OpenID
- mod_auth_openid
- Bindings for most languages: Perl, .NET, python
Questions?

rcritten@redhat.com