Shibboleth

SWITCH

Serving Swiss Universities

Chad La Joie (chad.lajoie@switch.ch)

Agenda

- What I will cover:
  - Introduction to Shibboleth (20 min)
  - Service Provider Overview (15 min)
  - SP/Application Integration (10 min)
  - Troubleshooting (5 min)
  - Q+A (15 min)

- What I will not cover:
  - Shibboleth Identity Provider
  - deployment trust models
  - configuration files
  - protocol messages

Evolution of Identity Management

- **Stone Age**
  Site-resident application maintains unique credential and identity information for each user
- **Bronze Age**
  Credentials are centralized (e.g. Kerberos, LDAP) but site-resident application maintains all user identity information
- **Iron Age**
  Credentials and core identity information is centralized and site-resident application maintains only app-specific data

- What's next?
  Allow users to use their "home" accounts with external applications - Federated Identities

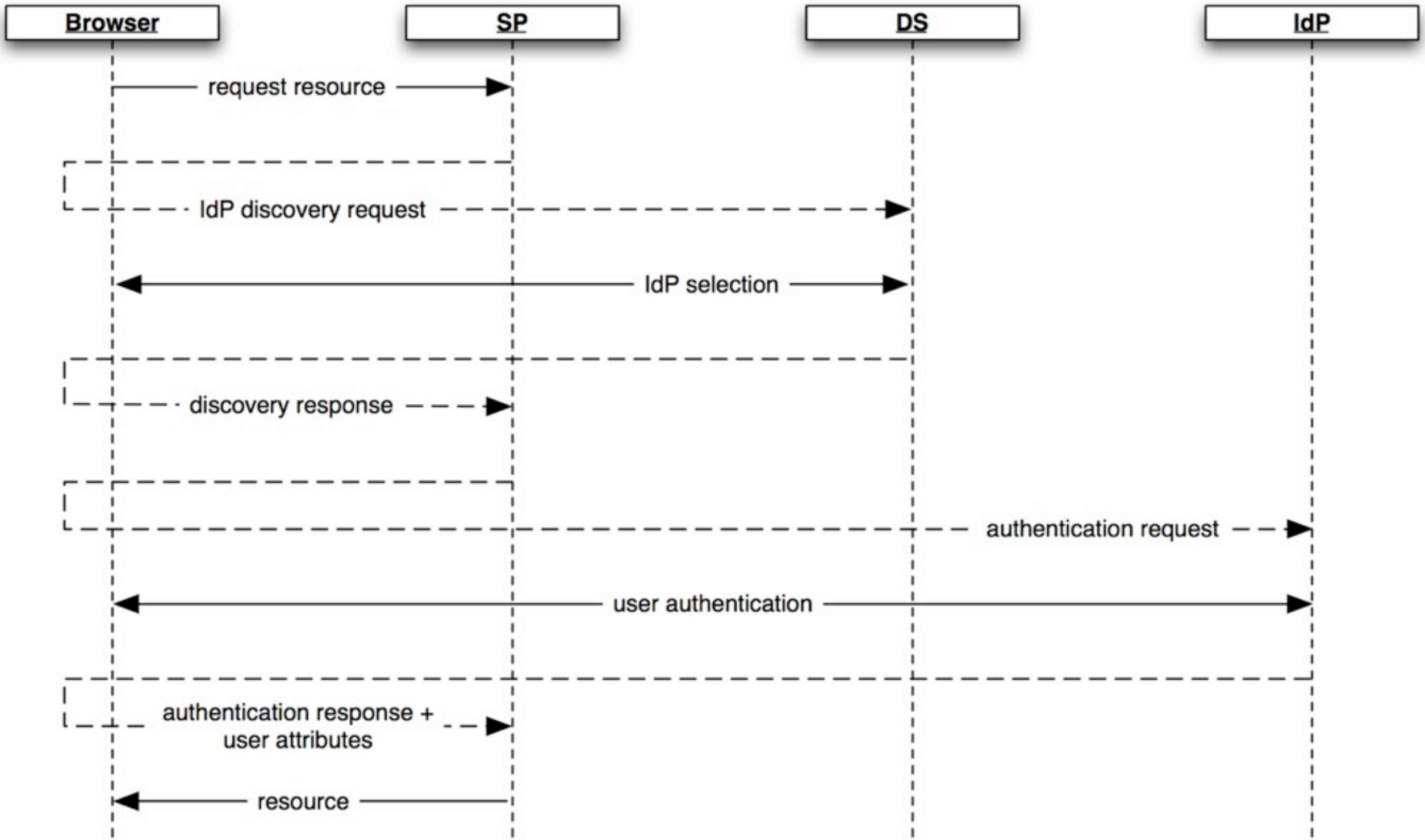
Shibboleth Components

- Identity Provider (IdP)
  - integrates with site's identity management systems
  - authenticates users
  - pushes identity information to protected services
- Service Provider (SP)
  - validate/process information from the IdP
  - makes information from IdP available to service
- Discovery Service (DS)
  - allows a user to tell an SP which IdP holds their account
- Metadata
  - an XML document; usually distributed via a well known URL
  - contains component endpoints, keys, contact info
  - serves as the foundation for technical trust

Shibboleth Protocols

- Security Assertion Markup Language (SAML)
  - **version 1** (5-Nov-02) - initial release, limited interoperability, limited use cases, no longer recommended for use
  - **version 2** (15-Mar-05) - current release, good interoperability, broader user cases, recommended for general use, Shibboleth 2 only
- Active Directory Federation Services (ADFS)
  - **version 1** - (2003) current release, terrible interoperability, suspect security, never supported within a Microsoft service
  - **version 2** - (2010?) in development, terrible interoperability (as of beta 2), support in “cloud” versions of Exchange and Sharepoint
- Others: eAuth, Liberty ID-WSF SSOS, MS Cardspace, OpenID v.2 (in development)
- All protocols currently focus on browser-based use cases
  - initial non-browser work in development

Shibboleth Flow



Shibboleth Flow: IdP Selection

Shibboleth Flow: User Authentication

What about Federations?

- Federations are social constructs
  - a group of organizations that agree to play by a common set of rules
- Shibboleth knows nothing about federations
  - it simply operates on a pile of metadata
- So what do federations do?
  - maintain metadata files (validation of registrants, publishing, etc.)
  - set out common policies that form behavioral trust
  - establish practices to help facilitate interoperability
  - technical support and training
- It is important to understand that federations are made up of member organizations/people not IdPs and SPs
  - the software doesn't know about federations
  - protocol messages do not occur "within" a federation

What Does It Do For Me?

- Reduces work
  - Authentication-related calls to Penn State's help desk dropped by 85% after they installed Shibboleth
- Provides current data
  - Studies of applications that maintain user data show that the **majority** of data is out of date. Are you "protecting" your app with stale data?
- Insulation from service compromises
  - In FIM data is pushed to services as needed. If those services are compromised the attacker can't get everyone's data.
- Minimize attack surface area
  - Only the IdP needs to be able to contact user data stores. All effort can be focused on securing this one connection instead of one (more) connection per service.

Other Benefits

- Users generally find the resulting single sign-on experience to be nicer than logging in numerous times.
- Usability-focused individuals like that the authentication process is consistent regardless of the service accessed.
- A properly maintained federation drastically simplifies the process of integrating new services.

Service Provider: Basics

- What does the Service Provider do?
  - initiates a Shibboleth authentication process
  - consumes the response from the IdP
  - provides IdP-supplied information to application
- What is the Service Provider?
  - a stateful daemon (shibd) that does most of the work
    - create, validate, and process protocol request messages
    - track/maintain sessions
    - cache information across requests
  - an IIS/iPlanet/Apache module that receives requests, pass them to shibd for processing, and puts the returned information in header/ environment variables
- Designed such that applications do not know about it
  - allows you to replace the SP with some other technology later

Service Provider: Metadata

- SP needs metadata for every IdP to which it communicates
- Loaded metadata - configured in `shibboleth2.xml`
  - may be loaded from a file or remote URL
  - remote files are cached locally and used if remote host is down
- Metadata Filters allow for metadata post-processing
  - always enable signature and `validUntil` filters
  - ensures the integrity and trustworthiness of the metadata
- You can get a reasonable start with an SP's metadata by access /Shibboleth.sso/Metadata
  - **NOTE:** it is the SP admin's job to make sure the metadata provided to others is correct and valid, the auto generation scripts provides a starting point, not a finished product

Service Provider: Session Initiation

- Types of Session Initiation (SI)
  - IdP initiated - IdP sends an unsolicited authentication response
  - required session - requires an SP session before viewing a resource, if no session is present then the login process begins
  - lazy session - user is free to browse around until a login session is explicitly started (e.g. by pressing a login button)
- Required Sessions
  - configured via normal web server mechanisms
    - e.g. `AuthType` and `Require` mod_auth commands in Apache
- Lazy Sessions
  - SI endpoint configured in SP then invoked via link/button
  - can configure SI endpoints specific to an IdP
    - allows user to bypass the discovery process
  - default /Shibboleth.sso/Login works in many cases

Service Provider: Attributes

- Attribute - a piece of information about the user
- Three steps to make attributes available to application:
- Attribute extraction - `attribute-mapping.xml`
  - decodes information in a protocol message in to a string value
  - determines request header/env variable names
- Attribute acceptance - `attribute-policy.xml`
  - whether the IdP is allowed to assert a given attribute/value
- Attribute export
  - places attribute/value in to CGI headers (IIS/iPlanet/Apache) or request environment variables (Apache)
  - **NOTE:** CGI headers maybe spoofed and most servers do little or nothing to prevent this
- Configuration files usually only need to be changed if custom attributes needs to be consumed

SP/Application Integration

1. Load one or more metadata sources to cover all the IdPs with which your application will work
2. Customize Session Initiator as appropriate
3. Point application login link/button to Session Initiator
4. Customize error pages as appropriate
5. Customize Attributes as appropriate
6. Use attributes, located in headers, in your app
   - Java: HttpServletRequest.getHeader(“foo”)
   - Perl: $ENV{“foo”}
   - PHP: $_SERVER[“foo”]
   - Python/Ruby - depends on framework used

SP/Application Integration: Issues

- Most applications expect to perform authentication
- Most applications expect all data to come from their own personal data source
- Both stem from an idea that technology never changes within the lifetime of an application
  - probably true for applications that only stick around for 1-2 years
- Addressing these issues helps the application:
  - easier to development
  - easier to deploy in various environments
  - “future proof” in terms of authentication and user data management
  - reduce overall surface area for bugs within the application

Troubleshooting

- Check the SP's log, most problems are caused by:
  - metadata failing to load (e.g. wrong URLs, expired metadata)
  - misconfigured keys (e.g. updated on file system but not in metadata)
  - mistyped entity IDs and endpoint URLs (e.g. http vs. https)
  - attributes being filtered out because of improper mapping/policy config
- Check the frequent errors page on the Shibboleth doc site
- Check the mailing list archives
- Ask on the Shibboleth user's list
  - include debug logs for a single request
  - state your environment and software versions
- If you're running an SP you should set up an IdP and gain experience with it