MIT Touchstone Configuration

IAP 2009

The basics of configuring a web application to use MIT Touchstone.

Bob Basch and Paul Hill

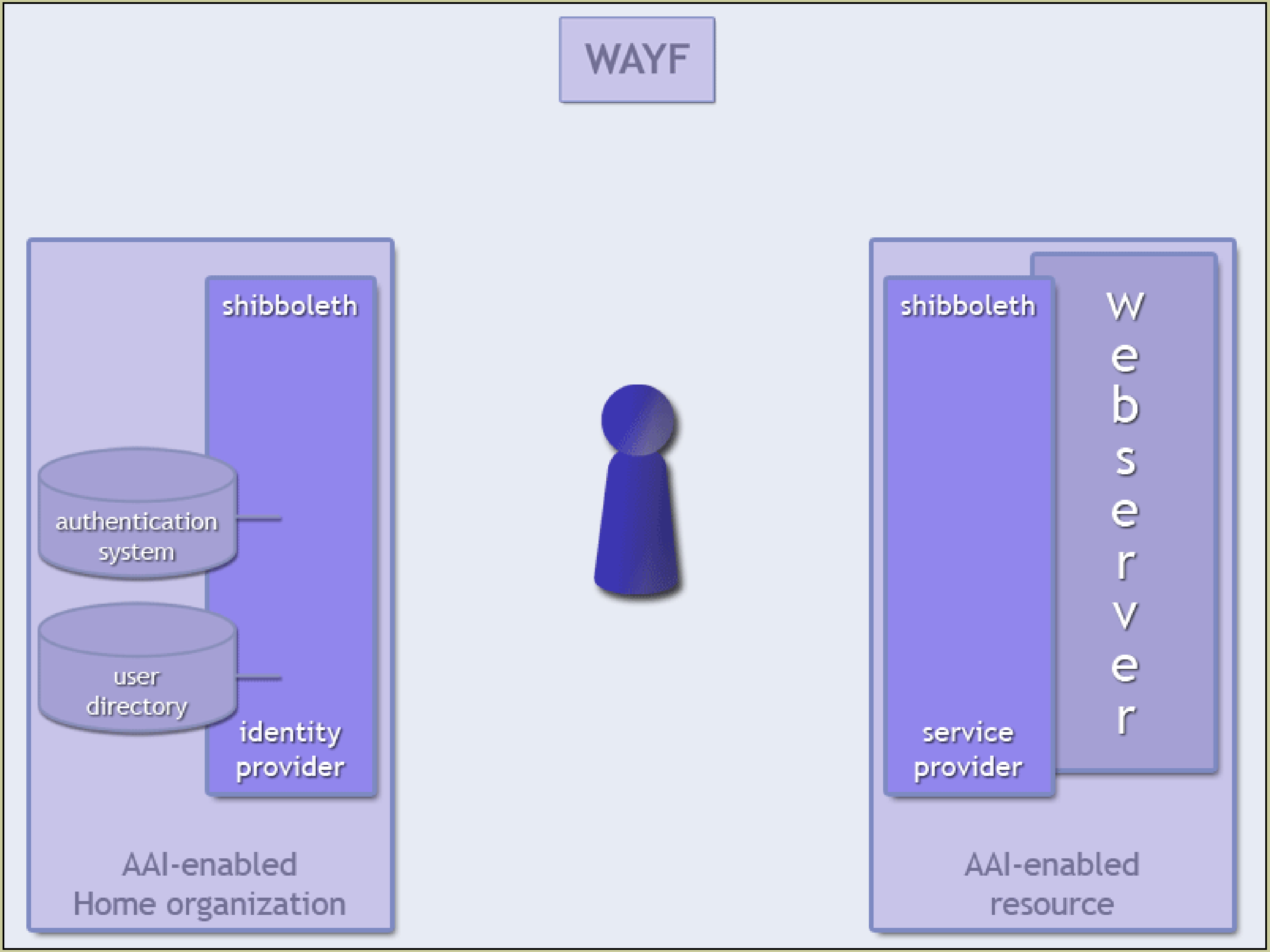
Information Services & Technology

Massachusetts Institute of Technology

Touchstone Overview

- Federated Authentication
- Identity Provider (IdP)
  - MIT
  - TouchstoneNetwork.net (aka CAMS)
- Shibboleth
- SAML
- InCommon
- http://mit.edu/touchstone

Overview of the interactions



MIT attribute release policies

- We release affiliation only, to applications that we don’t know about. (By default all InCommon SPs)
- We release affiliation, email, display name, EPPN (MIT username) to MIT applications that we know about.
  - We can discuss others on an application by application basis
- TouchstoneNetwork: email, EPPN (@touchstonenetwork.net), display name (first, last)

Assumptions for today’s talk

- Apache
- Internet2 Shibboleth distribution
- Focus on RHEL 5 as the OS platform

Alternatives that we’re not covering today

- IIS 6
- Java without an Apache or IIS front end
- SimpleSAMLphp - not currently supported
- OIOSAML - not currently supported
- Other OS versions: Solaris, MacOS, other Linux distributions

Components

We’re not covering basic installation today. There will be a hands-on installation session on Tuesday, January 20th, from 1:30-4:00pm

- Apache
- Shibd
- X.509 server certificates

- Getting started with installation and configuration:

https://wikis.mit.edu/confluence/display/TOUCHSTONE/Provisioning+Steps

Server Certificates

- SSL is used to authenticate and protect the communications between the SP and the IdP
- How to obtain a server certificate: https://wikis.mit.edu/confluence/display/WSWG/How+to+acquire+and+verify+a+M.I.T.+x509+Server+Certificate
- Please use all lower case for the hostname.
- Misconfiguration of the certificate is a leading cause of problems within the global Shibboleth community

Troubelshooting: certificate issue

- Check expiration date
- Do the processes have the necessary privileges to read the key and cert?
- Check log files

SSL

- Certificate may also be used to secure the connection between the browser and the web application
- At MIT we assume that you are using the same certificate for securing the browser session and the Shibboleth server to server connection.
- (Fewer certificates to manage or have expire)
- Applications in InCommon will have to manage an additional certificate

secure cookies

- Failure to protect the cookies can result in session hijacking
- The distributed shibboleth.xml does not set cookieProps to ensure that cookies are sent only for secure connections, because that could result in a redirect loop if accessing a non-https resource.
- cookies are protected by adding the cookieProps setting:

```
cookieProps="; path=/; secure"
```

Log files

https://spaces.internet2.edu/display/SHIB/LogFiles

- Change location of native.log from

  var/log/httpd/ to

  /var/log/shibboleth/httpd/

By editing etc/shibboleth/native.logger

- Set proper permissions on native.log (should be owned by Apache user)

Overview of the Shibboleth config files

- AAP.xml
- MIT-metadata.xml
- Other metadata.xml files
- Shibboleth.xml

AAP.xml

- MIT SP will rarely need to modify this file.
- Why might you modify it?
  - You might want to map an attribute to a different variable
    - Example: Fooling a SiteMinder compatible application

MIT-metadata.xml

- App maintainers don’t normally modify this file.
- Use a cron job to keep this up to date
  - /mit/touchstone/shibboleth/config/metadata/update-metadata.sh-example
  - http://mit.edu/touchstone/shibboleth/config/metadata/update-metadata.sh-example
- You must register you application with IS&T (or InCommon) and we maintain the file.

Meta Data to be provided

- Contact email address, (list, not personal account)
- The hostname
  - If multiple applications, each application name
- Organization (typically the DLC running the application)
- Organization’s URL

Display metadata files…

- See http://mit.edu/touchstone/config/shibboleth-sp/MIT-metadata.xml
- See http://wayf.incommonfederation.org/InCommon/InCommon-metadata.xml

Shibboleth.xml

- The local configuration file for each SP or IdP
- Large number of configuration options
- When using Apache, you can perform much of the configuration using Apache directives instead of modifying Shibboleth.xml directly

Session Management

- Shibboleth versus Application
- Shibboleth session lifetime versus inactivity timeout
- https://spaces.internet2.edu/display/SHIB/SessionManagement

Session management (2)

- Your application may want to manage its own session. This is especially preferable in a clustered environment, as the SP does not support clustered sessions itself.

Session Management (3)

- Shibboleth session timeouts are configured in shibboleth.xml's <Sessions> element.
- There are two timeout settings:
  - "lifetime" is the maximum lifetime of the session,
  - "timeout" is the idle session timeout, in seconds.
  - In the initial shibboleth.xml, these are set to 2 hours and 1 hour, respectively:

```
lifetime="7200" timeout="3600”
```

Session Management (4)

- After the session times out, the user will be bounced back to the IdP to create a new authenticated session. (If they still have a valid SSO cookie with the IdP, they will not have to authenticate again there).
- When the application manages its own session, and the Shibboleth session is only used at the initial entry point, the timeouts can be set small. You may then want to reduce the cacheTimeout setting in the global <MemorySessionCache> element to minimize the resources used to cache the sessions.
- Internet2 recommends setting the Shibboleth session lifetime to be the same as your application’s session lifetime

Session Management (5): IP address checking

- IP Address checking: There are two distinct settings:
  - "checkAddress"
  - "consistentAddress"

See https://spaces.internet2.edu/display/SHIB/AddressChecking

Session Management (6): checkAddress

- "checkAddress"
- set to "false" in the distributed shibboleth.xml
- ensures that the address of the client placed into the initial SAML assertion created by the IdP and the address of the client that delivers the assertion to the SP are the same.
- Set false initially because often clients may use different addresses when accessing different web servers.
- Proxy issue

Session Management (7): consistentAddress

- "consistentAddress"
- set to "true" in the distributed shibboleth.xml
- ensures that the client uses the same address when accessing a protected resource as when the SP session was established.
- This should always be set "true" to prevent session hijacking via cookie theft.

Session Management (8): SSL

- You should generally use SSL for all traffic with the SP session handler (leave handlerSSL set to "true").
- The distributed shibboleth.xml does not set cookieProps to ensure that cookies are sent only for secure connections, because that could result in a redirect loop if accessing a non-https resource.

Session Management (9): SSL (2)

- We recommend that all traffic accessing protected resources use https
- the ShibRedirectToSSL Apache directive is helpful here
- cookies are protected by adding the cookieProps setting:

```
cookieProps="; path=/; secure"
```

Session Management (10): handlerURL

- handlerURL
- “the gateway” to the resources you want to protect
- Each application must have an effectively unique handler location.
- The standard value is /Shibboleth.sso, i.e. with no hostname specified, so that it works for all vhosts.

SessionInitiator

- The <SessionInitiator> elements in shibboleth.xml define the locations used to initiate sessions, by redirecting to a WAYF or directly to a specific IdP's SSO endpoint.
- For automatic session setup, the first <SessionInitiator> element is the default
- to use a specific <SessionInitiator>, use the Apache ShibRequireSessionWith directive, or some applications provide a UI to make the selection

Shibboleth.xml SessionInitiator examples

```
<!-- This directs users to the MIT WAYF staging service. -->

<SessionInitiator id="MIT-WAYF-staging"
    Location="/WAYF/MIT-WAYF-staging"
    Binding="urn:mace:shibboleth:sp:1.3:SessionInit"
    wayfURL="https://idp-staging.mit.edu/WAYF/WAYF"
    wayfBinding="urn:mace:shibboleth:1.0:profiles:AuthnReque
    st"/>
```

Shibboleth.xml SessionInitiator examples (2)

```
<!-- This directs users to the MIT WAYF production service. -->

<SessionInitiator id="MIT-WAYF" Location="/WAYF/MIT-WAYF"
    Binding="urn:mace:shibboleth:sp:1.3:SessionInit"
    wayfURL="https://idp.mit.edu/WAYF/WAYF"
    wayfBinding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"/>
```

Shibboleth.xml SessionInitiator examples (3)

```
<!-- This directs users to the MIT core IdP's SSO service. -->

<SessionInitiator id="MIT" Location="/WAYF/mit.edu"
    Binding="urn:mace:shibboleth:sp:1.3:SessionInit"
    wayfURL="https://idp.mit.edu/shibboleth-idp/SSO"
    wayfBinding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"/>
```

Shibboleth.xml SessionInitiator examples (4)

```
<!-- This directs users to the MIT (core) staging IdP's SSO
    service. -->

<SessionInitiator id="MIT-staging" Location="/WAYF/idp-staging.mit.edu"
    Binding="urn:mace:shibboleth:sp:1.3:SessionInit"
    wayfURL="https://idp-staging.mit.edu/shibboleth-idp/SSO"
    wayfBinding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"/>
```

Shibboleth.xml, SessionInitiator, wayfURL

- The wayfURL is the redirection URL for the IdP or WAYF. The Location can be used in the "lazy sessions" case. The following query string parameters are used:
  - target the resource to direct back to later (or homeURL will be used)
  - acsIndex optional index of an ACS to use on the way back in
  - providerId optional direct invocation of a specific IdP

Lazy Sessions

"Shibbboleth also supports so-called lazy session establishment, in which the resource may be accessed without prior authentication. This means the application must be intelligent enough to determine whether authentication is necessary, and then construct the proper URL to initiate a browser redirect to request authentication; if the application determines none is necessary or uses other authorization mechanisms, then the request for authentication may not need to be triggered. This complex functionality is mostly useful to protect a single URL with different access mechanisms, or to require authenticated access only in instances where the application deems it necessary."

Using Apache to protect content

- Apache
  - Using Apache to protect content, not modifying shibbboleth.xml
  - <location></location> examples
  - Lazy sessions versus required

Apache, global configuration

- You should always set ServerName explicitly in Apache, and should always set UseCanonicalName to "On", to ensure that the name used matches the <Host> element in shibboleth.xml.
- The apache2[2].config files in etc/shibboleth contain the basic set of directives for your Apache version.
- Make sure that the Shibboleth module is loaded, e.g.:

```
LoadModule mod_shib /usr/libexec/mod_shib_22.so
```

Apache, global configuration (2)

- Define the configuration file path, and the schema directory:

```
ShibSchemaDir /usr/share/xml/shibboleth
ShibConfig /etc/shibboleth/shibboleth.xml
```

- These are used for the logoLocation and styleSheet paths in the error templates (see the <Errors> element in shibboleth.xml):

Example

```
<IfModule mod_alias.c>
   <Location /shibboleth-sp>
         Allow from all
   </Location>
   Alias /shibboleth-sp/main.css /usr/doc/shibboleth/main.css
   Alias /shibboleth-sp/logo.jpg /usr/doc/shibboleth/logo.jpg
</IfModule>
```

Apache, protecting content

Content may be protected either via blocks in Apache configuration files (e.g. httpd.conf) or .htaccess files.

The basic set of Apache directives for protecting content:

```
<Location /secure>
  AuthType shibboleth
  ShibRequireSession On
  require valid-user
</Location>
```

Apache, protecting content

This will require an authenticated session to be established. Note that the "require valid-user" directive means that any user who authenticates at an IdP will be accepted, even if the SP receives no attributes, including the user name. (In this case it is expected that the application will do proper checking of REMOTE_USER).

Apache protected content

Protected content should use SSL. To redirect non-SSL GET or HEAD requests automatically, include the ShibRedirectToSSL directive, e.g.:

```
ShibRedirectToSSL 443
```

You should also set your cookies to be secure in shibboleth.xml

Adding granularity in .htaccess

For cases where more explicit checking is needed outside of the application, Shibboleth extends the require directive to support checking against a regular expression, e.g.:

```
require user ~ ^.+@mit.edu$
```

This would require that the authenticated user name ends in "@mit.edu", i.e. was authenticated against an MIT IdP. (Though this does not preclude the user from authenticating against our staging IdP; to explicitly check that the user authenticated against a particular IdP, the application should check the HTTP_SHIB_IDENTITY_PROVIDER variable).

Doing this via AAP.xml instead

Similar functionality may be used to add a rule for matching an attribute alias, as defined in AAP.xml, e.g.:

```
require affiliation ~ ^student@mit.edu$
```

In general, it is usually preferable for the application to perform these checks, where possible.

Apache, selecting the session initiator

By default, Shibboleth will redirect the user to the location defined for the first <SessionInitiator> element in shibboleth.xml. To redirect to a specific WAYF or IdP, use the ShibRequireSessionWith directive. For example:

```
ShibRequireSessionWith MIT
```

will redirect to the MIT core IdP (if using the shibboleth.xml generated by gen-shib; the value specified here must match the value of the SessionInitiator id attribute).

Apache, lazy sessions

For "lazy" sessions, i.e. where the application triggers the establishment of a Shibboleth session, use a directive like this to ensure that the request always goes through the Shibboleth module, to prevent header spoofing.

```
<Location />

  AuthType shibboleth

  require shibboleth

  ShibRequireSession Off

</Location>
```

Apache, lazy sessions (2)

The "require shibboleth" directive says that if the expected session cookie exists, the session will be validated and any cached attributes will be exported. If the session does not already exist, though, no session is requested, and control is passed to the resource.

(An example where this is required is when using the Drupal Shibboleth module).

Protecting Content Documentation

- Detailed explanation of how to protect resources using Shibboleth is described in: https://spaces.internet2.edu/display/SHIB/SPProtectionConfig

Apache, Tomcat and AJP connector

- When using Tomcat we recommend using it with Apache and mod_jk, or mod_proxy_ajp
- There are several factors to consider when choosing which mechanism to use, the mod_proxy_ajp method is easiest to setup for those with apache 2.2+ but can lack some options. There is a good comparison of the two methods (mod_jk and mod_proxy) posted by one of the tomcat lead developers at http://blogs.jboss.com/blog/mturk/2007/07/16/Comparing_mod_proxy_and_mod_jk.txt

Multiple applications on the same Apache server

The main reasons for doing so:

1. One application always requires authentication, the other will use lazy sessions
2. Different attribute release policies for the applications

Multiple applications on the same Apache server (2)

- Map each application to a different applicationID (RequestMap element within Shibboleth.xml)
  - You can define a separate applicationId for any <Host> or <Path> element.
- Make the new applicationId identify itself with a separate providerId (Applications element)
- Each application has a providerID which must be known to the IdP (via the metadata.xml file)

Multiple applications on the same Apache server (3)

- Internet2 documentation on this subject can be found at:

https://spaces.internet2.edu/display/SHIB/AddSeparateApplication

Multiple applications on the same Apache server: Each application SHOULD be on a separate virtual host

Why we recommend using vhosts:

Failing to do this will not provide any real separation of applications because any resource in one application can generally trick the client into supplying the cookie issued by another.

From https://spaces.internet2.edu/display/SHIB/AddSeparateApplication

Restart the applications independently when application specific configuration information has changed

Easier to maintain the shibboleth.xml file

Multiple applications on the same Apache server

- Why you might use non standard port numbers instead
- Plan Ahead!
- ApplicationID
- ProviderID
- Don't forget to get registered with the IdP

Adding your application to InCommon:

- Implications
  - Other sites can authenticate
- issues
  - Need a certificate from InCommon
  - Need to register application with InCommon
- Steps
  - Docs will be available via http://mit.edu/touchstone