Integrating Web Applications with Shibboleth

Application Authentication Done Right

July 11, 2016
Eric Goodman, UCOP IAM Architect
Jeffrey Crawford, UCSC Application Admin
What is Shibboleth?

- Shibboleth
  - An open-source, higher-ed funded application
  - Implements SAML protocol to support federated authentication
  - Authenticates users of web apps by leveraging existing user accounts

- So what is SAML?
  - “Security Assertion Markup Language”
  - Protocol (standard), NOT an application
  - Communicates Authentication information
    - Describes who(ish), how and when the user logged in
  - XML-based
  - Secure
    - Uses asymmetric key cryptography for message signing/encryption

- SAML is the UC Standard for cross-campus authentication
UC Standards?

• UC IT Architecture Committee (ITAC)
  – Working sub-committee to the UC IT Leadership Council (ITLC)
  – Mission: “establish the enterprise architecture and technology infrastructure necessary to promote and support interoperability and sharing of IT solutions among and between campuses”
    • Foundation for Collaboration on Technology Enabled UC Solutions, Pillar II
  – Develops principles, standards and practices of Enterprise Architecture across UC
  – Facilitates knowledge sharing and collaboration across campuses.
  – Supports implementations of UC-wide initiatives through consultation and maintenance of EA Body of Knowledge
More on ITAC, Standards, Etc.

• Enterprise Architecture Body of Knowledge (EA BoK)
  – Repository of IT principles, standards, guidelines and other EA Artifacts (EAAs)
  – Reviewed by campus communities and adopted by CIOs
  – SAML for app authentication (EAA-006) is one of the standards
    • Formally adopted at more than half of campuses
    • De-facto standard via UCTrust (i.e., all campuses support this)

• UCTrust
  – Subcommittee of ITAC
    • Mission: “provide input on design and implementation of identity management solutions for the UC system and to foster collaboration on IAM solutions among the campuses”
  – Members provide direct support of SSO (Shibboleth) integration
Federated Authentication and UC Enterprise Architecture Standards
More info

• ITAC
  – UCCSC Presentation and overview
    • 3:00 Monday, Porter D246
  – Website
    • https://spaces.ais.ucla.edu/display/ucitag/

• UCTrust
  – Website
    • https://spaces.ais.ucla.edu/display/uctrustwg/
What is Shibboleth?

• Shibboleth
  – An open-source, higher-ed funded application
  – Implements SAML protocols to support federated authentication

• So what is SAML?
  – “Security Assertion Markup Language”
  – Protocol (standard), NOT an application
  – Communicates Authentication information
    • Describes who(ish), how and when the user logged in
  – XML-based
  – Secure
    • Uses asymmetric key cryptography for message signing/encryption

• SAML is the UC Standard for cross-campus authentication
Why “Shibboleth”?

…and it was so, that when those Ephraimites which were escaped said, Let me go over; that the men of Gilead said unto him, Art thou an Ephraimite? If he said, Nay; Then said they unto him, Say now Shibboleth: and he said Sibboleth: for he could not frame to pronounce it right. Then they took him, and slew him at the passages of Jordan: and there fell at that time of the Ephraimites forty and two thousand…

— Judges 12:5–6, King James Bible
Comparison of Authn Approaches

• Local authentication

• Pass-thru (proxy) authentication

• Authentication as a service
Local Authentication

• Use application-specific passwords
Local Authentication

User and User's Computer

Web Application #1
- Verify
  - Username: ericg
  - Password: mybadpwd

Web Application #2
- Verify
  - Username: egoodman
  - Password: myotherbadpwd

User types in "ericg" + "mybadpwd" for Web Application #1.
User types in "egoodman" + "myotherbadpwd" for Web Application #2.
Local Authentication - Scaling

Passwords Everywhere!
Local Authentication

• Pros
  – Flexibility
    • Different usernames and passwords for each site
  – Simple to set up

• Cons
  – Usability
    • Different usernames and passwords for each site
    • Doesn’t integrate with anything else
    • Password changes are per-application
  – App must support password reset/I forgot my password
  – Security
    • Strong risk that users will reuse passwords
    • Passwords are confidential data and require extra security!
Pass-thru (proxy) Authentication

- Externalizes authentication
- Application impersonates user

“Verifier” can be LDAP, AD, Kerberos, etc.
Pass-thru Authentication
Pass-thru Authentication - Scaling

Passwords transmitted everywhere!
Pass-thru Authentication

• Pros
  – Same username password at each site
  – Single database for account/password changes

• Cons
  – Each app directly handles/transmits passwords
  – Trains users to enter UC password on any site
    • User has no way to validate website
  – Application is the user
    • AuthN service can’t distinguish you and application
Federated Authentication

- Authentication as a service
- Uses the SAML protocol
Federated Authentication - Scaling
Federated Authentication

• Pros
  – A single, trusted application handles the passwords
  – Users always enter UC passwords on same website
  – Application sees approved user info, does not act “as user”
  – Authentication process handled centrally
    • Can leverage central services; multi-factor authentication, expired accounts
  – Single Sign On
  – Supports logins from multiple campuses without requiring new accounts
  – Easy to “Export”
    • E.g., safe to install and run in cloud environments (like AWS)

• Cons
  – Largely Web-Only
  – Complexity of initial install
  – Onboarding process at UC is currently “clunky”
  – Vendor adoption
Federated Authentication: SAML and Shibboleth

How it Works
Shibboleth components

• Identity Provider (IdP)
  – Run by the campus (organization)
  – Performs authentication (login)
  – Provides verifiable attributes describing user

• Service Provider (SP)
  – Run at the application level (“protected resource”)
  – Communicates with and validates info from IdP
Things to get used to with SAML

- **IdP and SP don’t directly communicate**
  - AuthN info is shared via SAML messages
  - SAML messages are carried by user’s browser
    - Encryption and signing are important!

- **IdP does not control application session**
  - Tells SP about the user being authenticated
  - After login/authentication is complete, IdP is done
    - SP session is managed locally
  - Means that “Logout” is a whole different animal

- **SP does not control login process**
  - Asks IdP to do authentication
  - AuthN UI/flow is managed by the IdP
  - The IdP provides the user data (including username)
    - SP must use/map the IdP’s data for local profiles
Demo login (if time)
SAML Technical Notes

- Shibboleth SP software runs as a separate daemon/service
  - SP software maintains its own session
    - Application/web server can leverage session, but map to its own env
  - SP software translates SAML into http headers or web variables
  - Application reads SAML attributes to identify user
    - \$_SERVER["attribute_name"];
    - request.getHeader("attribute_name");
  - App can register users “on the fly” using this info
    - i.e., Create profile dynamically using IdP-provided information

- Other notes
  - Apps built from scratch are easy to integrate with Shibboleth
  - Third party apps can be more difficult depending on architecture
  - Applications that read REMOTE_USER generally integrate well
  - j2ee applications based on tomcat? use AJP if possible
### SAML Attributes

<table>
<thead>
<tr>
<th>eduPerson Attributes (InCommon/Internet2)</th>
<th>UCTrust Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>eduPersonAffiliation</td>
<td>UCNetID</td>
</tr>
<tr>
<td>eduPersonEntitlement</td>
<td>UCTrustAssurance</td>
</tr>
<tr>
<td>eduPersonNickname</td>
<td>UCCampusEmployeeID</td>
</tr>
<tr>
<td>eduPersonOrgDN</td>
<td>UCTrustShortCampusID (deprecated)</td>
</tr>
<tr>
<td>eduPersonOrgUnitDN</td>
<td>UCPathEmplid</td>
</tr>
<tr>
<td>eduPersonPrimaryAffiliation</td>
<td>UCCampusStudentId</td>
</tr>
<tr>
<td>eduPersonPrimaryOrgUnitDN</td>
<td>UCEmployeeStatus???</td>
</tr>
<tr>
<td>eduPersonPrincipalName</td>
<td>UCStudentGradeLevel???</td>
</tr>
<tr>
<td>eduPersonScopedAffiliation</td>
<td></td>
</tr>
<tr>
<td>eduPersonTargetedID</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Local Attributes (campus specific)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CruzID (UCSC)</td>
</tr>
<tr>
<td>FacultySystemID (hypothetical)</td>
</tr>
</tbody>
</table>
Using SAML in an App

• Install Shibboleth SP software (shibd)
  – Official RPMs/Installers
    • RHEL, CentOS, SUSE, Windows
  – Unofficial
    • (Li/U)nux systems, MacOS (MacPorts), Java Servlets

• Configure web server
  – Apache
    • Load module (mod_shib)
    • Protect content using normal “Location” directives
  – IIS
    • Enable ISAPI filter (isapi_shib.dll)
    • Protect in “shibboleth2.xml” using shibboleth config syntax
Using SAML in an App

- Configure Shibboleth Software
  - shibboleth2.xml
    - Controls Shibboleth options (config, basic settings)
    - Protection directives for IIS must go here
      - Uses a Shib-specific syntax
  - attribute-map.xml
    - Maps SAML attributes to variables/headers
  - Protection directives (Apache only)
    - Via standard <Location /> style configs
Configuring Shibboleth

• Set entityID in ApplicationDefaults
  
  ```xml
  <ApplicationDefaults
    entityID="https://idm-test-sp.ucsc.edu/shibboleth"
    REMOTE_USER="eppn persistent-id targeted-id">
  ```

• For https, change this in Sessions section
  
  ```xml
  <Sessions lifetime="28800" timeout="3600" relayState="ss:mem"
    checkAddress="false" handlerSSL="true" cookieProps="https">
  ```
Configuring Shibboleth (cont)

• Load Metadata (the config. info for campus ldPs)
  <MetadataProvider type="XML"
      uri="http://md.incommon.org/InCommon/InCommon-metadata.xml"
      backingFilePath="incommon-metadata.xml"
      reloadInterval="7200">
  </MetadataProvider>

• Identify login service?
  <SSO discoveryProtocol="SAMLDS"
       discoveryURL="https://wayf.incommonfederation.org/DS/WAYF">
      SAML2  SAML1
  </SSO>
Using SAML in an App

• Test with local campus IdP
  – May require loading local campus IdP configuration
  – May require configuration of campus IdP (by local IdM team)
  – Process varies, work with your campus contact:
    https://spaces.ais.ucla.edu/display/uctrustwg/UCTrust+Campus+Contacts

• Register application with InCommon
  – InCommon provides a registry of verified SP/IdP config info
    • Called Metadata
  – Shibboleth automatically loads/updates configs from InCommon
  – Work with campus contact to list your app in InCommon

• Work with campus contact to integrate with other campuses
  – Request appropriate attributes
Vendors and SAML

• SAML support is frequently minimal
  – Vendors don’t always use Shibboleth (the app)
  – Vendors may require manual configuration of IdP info
    • Rather than Shib’s auto config via InCommon metadata
  – Some vendors are limited to one campus (IdP)
  – “SAML Proxy” service may help
    • Available via UCOP for UC applications

• “SSO” does not always mean “SAML”
  – Can mean “we integrate with AD (or LDAP)”
    • I.e., Pass-thru or proxy authentication
    • Not what we use at UC for systemwide apps
  – Include SAML support in initial plans and RFPs
Question & Answer
For further questions

• Eric Goodman, UCOP IAM Architect
  – eric.goodman@ucop.edu
  – 510-587-6308

• Jeffrey Crawford, UCSC App Admin
  – jeffreyc@ucsc.edu

• Campus IAM contacts
  – https://spaces.ais.ucla.edu/display/uctrustwg/UCTrust+Campus+Contacts