

SHIBBOLETH AND SAML: AT LAST, A VIABLE GLOBAL STANDARD FOR RESOURCE ACCESS MANAGEMENT

John Paschoud

New Review of Information Networking, Vol. 10, No. 2, 2004

Presentation by Stuart Miller

Missouri S&T

CpE 5420



Introduction

- > Internet designed to be easily accessible and available to everyone
 - BERNERS-LEE, T. 1999. Weaving the Web. London: Orion Publishing
- > Internet contains no build-in standard for restricting access
- > Restricted access beneficial for
 - Communities of registered users (forums, research groups)
 - University resources
 - Material in libraries, journals, etc.

Requirements of a Resource Access Management System

Clifford Lynch (ed.) 1998. A White Paper on Authentication and Access Management Issues in Cross-Organizational Use of Networked Information Resources; Coalition for Networked Information; <http://www.cni.org/projects/authentication/authentication-wp.html>

- > **Simplicity**
 - Leads to easy adoption and acceptance
- > **Privacy**
 - Must protect the privacy of users from detailed tracking and disclosure of use
- > **Good faith**
 - Users and providers need to mutually depend on the system
 - All must be able to trust that it can protect against abuse
- > **Trusted intermediaries**
 - Often required to provide access to both users and providers. System must accommodate them
- > **Reasonable terms**
 - License agreements must not be overly tight
 - System must not impose unreasonable restrictions
 - System must not discriminate against particular user groups

Current Solutions

(at time of publishing in 2004)

> Common Shared 'Secret' Passwords

- Single secret password, shared by everyone
- Unlikely to stay secret for long
- Cannot be changed without communication to entire community

> Registration of Individual Users

- Requires some kind of contractual agreement with user (business client, student enrollment, etc.)
- Undesirable to give external applications access to user information
- Need to obscure user identity

> IP Address Restriction

- Only works for as long as IP addresses are static
- Anyone with physical access to that IP address can override this
- Doesn't account for remote or distance users

Current Solutions (continued)

(at time of publishing in 2004)

> IP Address Restriction, with Authenticated Proxy-servers

- Can be technically challenging to implement
- Complicated for end-users
- Must re-configure for each resource being accessed
- Unlikely to stay secret for long
- Cannot be changed without communication to entire community

> Athens System

- Eduserv Athens Service: <http://www.athensams.net/>
- Outsources management of usernames and passwords to central service
- Need to provide managed lists of which users have access to which resources
- Requires an annual fee
- Completely reliant on third party for ALL security
- Now known as OpenAthens and supports SAML, operating in a similar manner to Shbboleth

Principles of Shibboleth

> Functional separation

- Authentication – undertaken by host organization with access to privileged information
- Authorization – undertaken by resource provider which managed access

> Architectural Aspects

- Federated administration – A trust fabric must exist between all users of a Shibboleth Federation
- Access control based on attributes – Shibboleth users possess attributes which make access control decisions
- Active management of privacy – Shibboleth only released specified attributes to resource providers. By default, the only information provided is “member of community”
- Multiple, scalable trusts – Shibboleth can specify a different trust relationship with each individual party.
- A standard (yet extensible) attribute vocabulary - Shibboleth has defined a standard set of attributes, which can be expanded upon.

Shibboleth is NOT...

- > A method of authentication unto itself
 - User authentication is still done by an organization's internal mechanisms
 - Usually some type of “Single Sign-On” solution
- > Passwords are NEVER transmitted to resource providers
 - Not even in encrypted form
 - Can accommodate any advanced authentication (such as smartcards or biometrics) because this is all external to Shibboleth

How Shibboleth Works, Part I

- > The resource provider (ReP): The entity hosting the service that the user wants to access
 - SHIRE (SHibboleth Indexical Reference Establisher)
 - SHAR (SHibboleth Attribute Requester).
- > The identity provider (IdP): The home organization with which the user is registered. Provides initial authentication
 - HS (Handle Server)
 - AA (Attribute Authority).
- > The 'where are you from' (WAYF) service: Identifies where the user is from and which IdP should be used

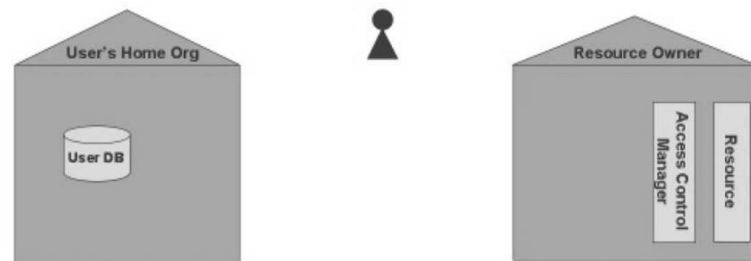


FIGURE 1

User, home organization (IdP) and resource owner (ReP)

How Shibboleth Works, Part II

1. Using her browser, Alice connects to the web-based resource.
2. Since the web server detects no established session for Alice, the server hands her request over to SHIRE, which redirects Alice's web browser to the WAYF server typically run by the federation of which the ReP is a member.
3. The WAYF server presents Alice a web page from which she selects the name of her home organization.

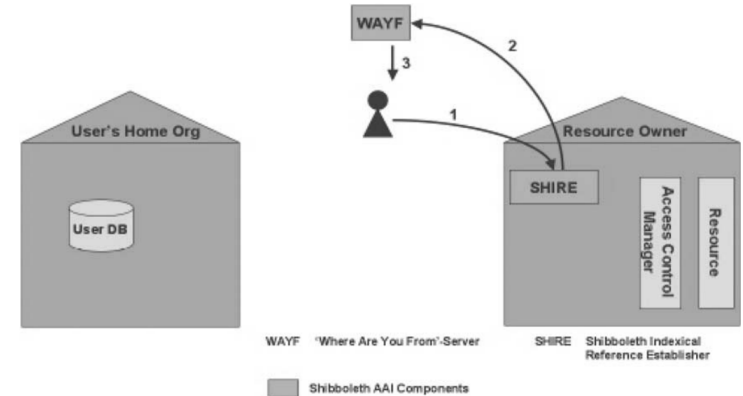


FIGURE 2

User connects to resource owner, and is redirected to WAYF

How Shibboleth Works, Part III

4. When Alice selects her home organization, her browser returns the selection to the WAYF.
5. The WAYF then redirects her web browser to the handle server (HS) of her home organization.
6. From the handle server, Alice gets a web login screen of her university, well known to her since she uses the web login already for various web resources offered by her university.

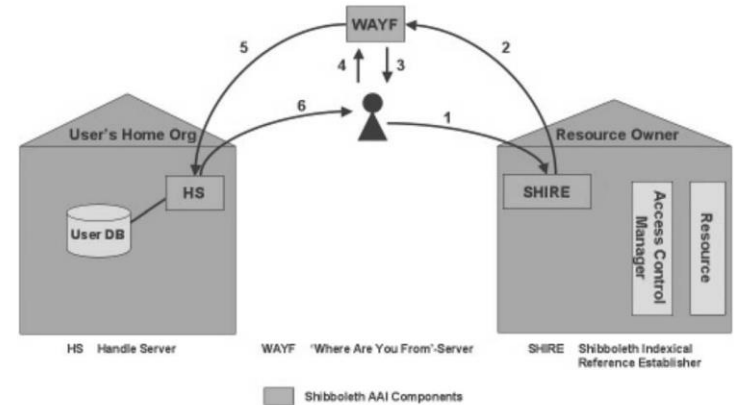


FIGURE 3

User selects her home organization and authenticates there

How Shibboleth Works, Part IV

7. Alice provides her credentials (e.g. username and password) to the handle server of her home organization via an existing and familiar web SSO (Single Sign-On) interface.
8. Provided the credentials are correct, the handle server generates an opaque and digitally signed handle on behalf of Alice, which is sent to the resource Alice wants to connect to by another web browser redirection.
- (This step is completely invisible for Alice, since it is a server to server communication in the background between the Shibboleth components at the ReP and IdP.)
- On the resource side, the Handle received gets passed to the SHAR (Shibboleth Attribute Requestor) component.

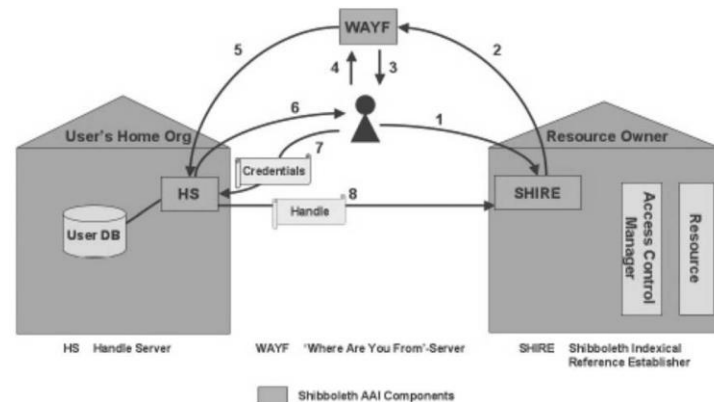


FIGURE 4

Authenticated user is redirected back to resource owner

How Shibboleth Works, Part V

- The SHAR then sends it via a secure HTTP connection to the attribute authority (AA) at the home organization that generated that handle.

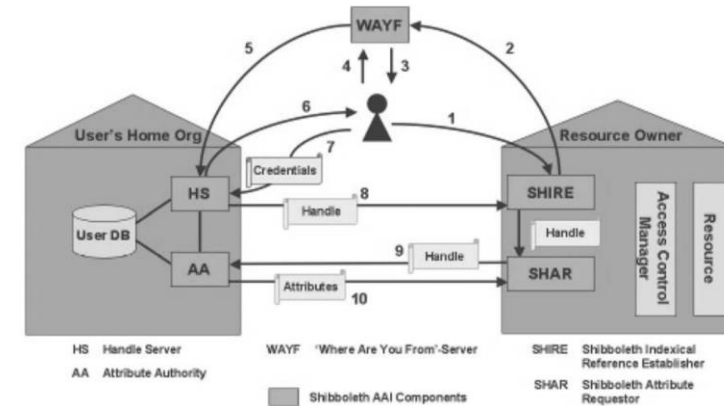


FIGURE 5

SHAR fetches user's attributes on behalf of the resource owner

How Shibboleth Works, Part VI

10. The AA verifies the handle and its validity internally with the handle server.

- If valid, the AA checks out which attributes it may release to the resource based on the attribute release policy (ARP) of Alice regarding the resource.
- The AA sends the attributes allowed to release, digitally signed, to the SHAR.
- Finally, the SHAR passes the attributes received to the access control manager which then, according to its configuration, authorizes the access for Alice based on the set of attributes provided.
- If a resource requires information about the user for functional purposes (such as personalization), the access control manager can request and pass these attributes to the resource.

> After this the session is established and Shibboleth is no longer involved

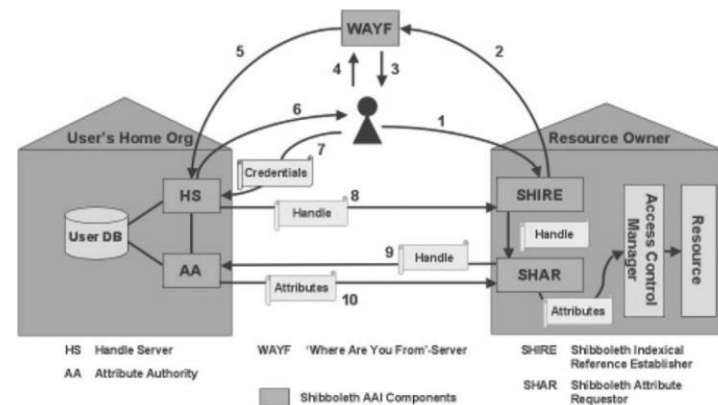


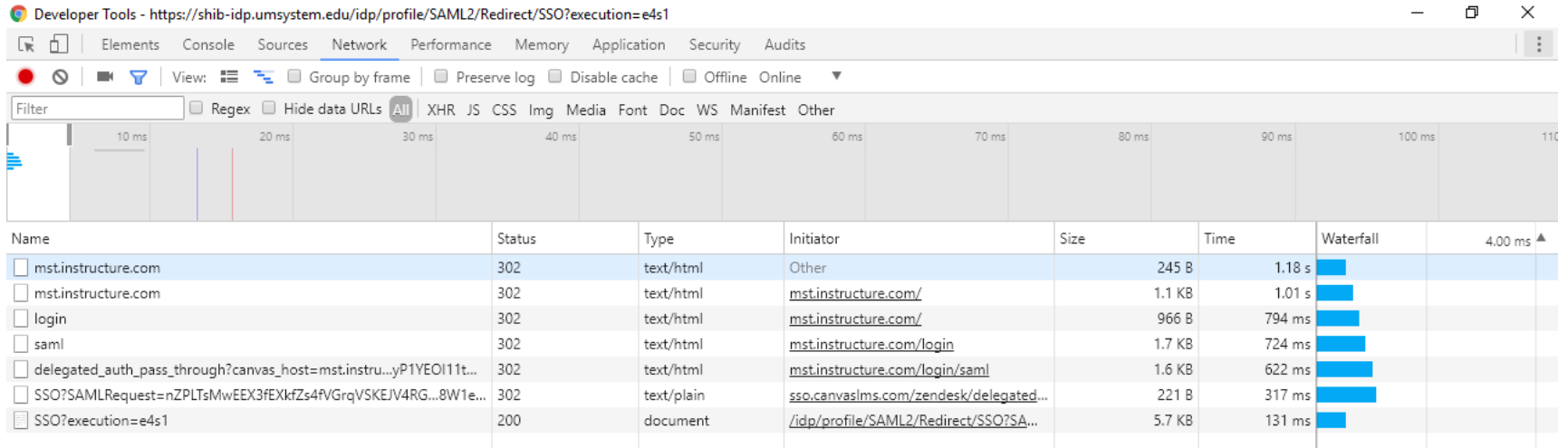
FIGURE 6

Access control manager decides on authorizing user's access

How Shibboleth Works at S&T - Overview

- > Example with logging in to Canvas
 - Canvas is hosted externally (Blackboard used to be hosted by S&T)
 - Owned by third party company, Instructure
 - Uses Shibboleth
- > Can log in at <http://mst.instructure.com/>

How Shibboleth Works at S&T – IdP



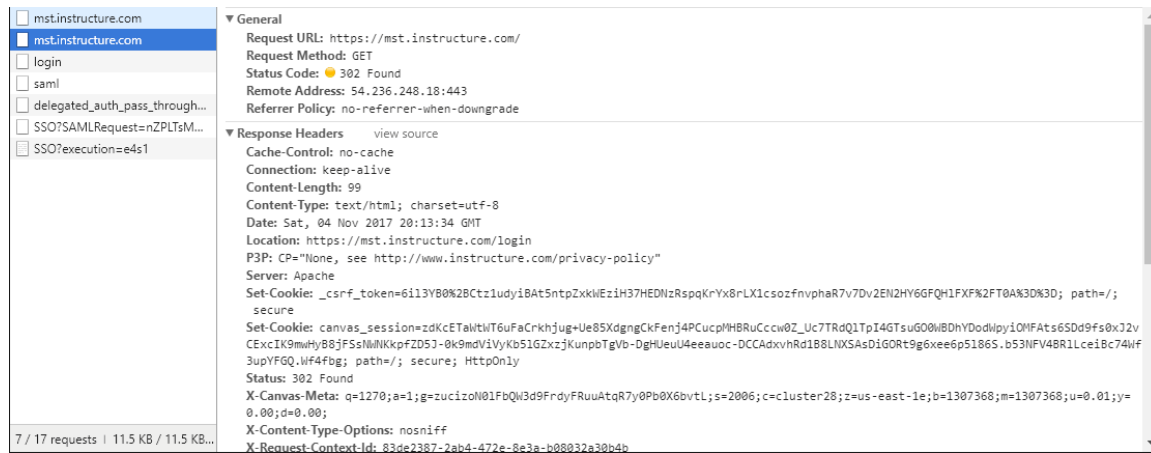
How Shibboleth Works at S&T – IdP

- > First few requests are simple redirects
 - Instructure's sever recognizes mst.instructure.com and know what login configuration to return
 - We get redirected through a series of URLs for our configured login procedure



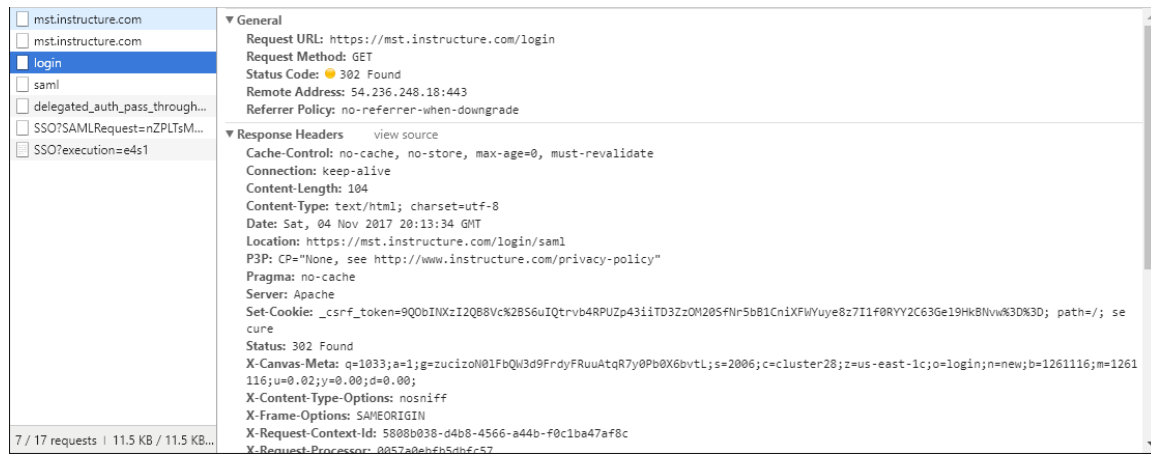
How Shibboleth Works at S&T – IdP

- > First few requests are simple redirects
 - Instructure's sever recognizes mst.instructure.com and know what login configuration to return
 - We get redirected through a series of URLs for our configured login procedure



How Shibboleth Works at S&T – IdP

- > First few requests are simple redirects
 - Instructure's sever recognizes mst.instructure.com and know what login configuration to return thanks to SHIRE
 - We get redirected through a series of URLs for our configured login procedure



How Shibboleth Works at S&T – IdP

- > First few requests are simple redirects
 - Instructure's sever recognizes mst.instructure.com and know what login configuration to return thanks to SHIRE
 - We get redirected through a series of URLs for our configured login procedure

mst.instructure.com

- mst.instructure.com
- login
- saml
- delegated_auth_pass_through...
- SSO?SAMLRequest=nZPLTsM...
- SSO?execution=e4s1

General

- Request URL: <https://mst.instructure.com/login/saml>
- Request Method: GET
- Status Code: 302 Found
- Remote Address: 54.236.248.18:443
- Referrer Policy: no-referrer-when-downgrade

Response Headers

- Cache-Control: no-cache, no-store, max-age=0, must-revalidate
- Connection: keep-alive
- Content-Length: 856
- Content-Type: text/html; charset=utf-8
- Date: Sat, 04 Nov 2017 20:13:34 GMT
- Location: https://sso.canvaslms.com/zendesk/delegated_auth_pass_through?canvas_host=mst.instructure.com&target=https%3A%2F%2Fshib-idp.umsystem.edu%2Fidp%2Fprofile%2FSAML2%2FRedirect%2FSAMLRequest%3DnZPLTsMwEEX3fEXkFZs4fVGrqVSKJEJv4RG1gwc51pmApsYnnAuXvsUtB1YAuurI0nsy90Xh9QV1XjZi19GKw8NoCuRStK4Nid5Gx1h1hJMoURtAgpRYzW5vRNpNROMsMWUrf10uM7Z003U%252FPeethA%252Fg95Ycj4epaVM5Dr1ZcLPR4NRORwOJWfRIzju1mThj%252Ffi7awMEjSkC81fNThvJP0iZQRvCd6%252F5cW5XutC21KbZ6PG1t%252FNaG4Lq8k9%252BvChbNEMGRF51bg20NbgXuTst4hN5k71WoQRHhWJXeY0uVdQ66Cpbx5V911Y00Nn0Eg6x8%252B80813I7%252Bv2T1T0%252FCMT5qSTN%252BEBgejb52s6dh7m4zG211ccp27myrp20vDtUdNnZ7FoFOM1QgyHPPrans%252B9yB3M1YdwyEqvzb14%252FzFyq3GXkwybYnpSpua0b6T5GmBhKvrgfzh4Xnm659icsoyjbUqoMnqXc3%252B8W1eG8ILyP1YE0I11tN%252FiX35%252BGP2Jw8OKfz%252B86Sc%253D
- P3P: CP="None, see <http://www.instructure.com/privacy-policy>"
- Pragma: no-cache
- Server: Apache
- Set-Cookie: _csrf_token=4DIwLfhJb21FPXLrrdIe6%2BLS1hgoqrj374nw0DAgZn2heFhnaAo5AzbMLPisHma1JumTXHj%2Bp%2BGuKxkU8MTDw%30%3D; path=/; secure
- Status: 302 Found

7 / 17 requests | 11.5 KB / 11.5 KB...

How Shibboleth Works at S&T – IdP

- > First few requests are simple redirects
 - Instructure's sever recognizes mst.instructure.com and know what login configuration to return thanks to SHIRE
 - We get redirected through a series of URLs for our configured login procedure



How Shibboleth Works at S&T – IdP

- > Here is the actual SAML request
- > Note that here, a cookie to identify our session is set.
 - “JSESSIONID”

General

Request URL: <https://shib-idp.umsystem.edu/idp/profile/SAML2/Redirect/SSO?YNNaUxvsUtB1YauurI0nsy90XW9QV1Xj2i19GKH8NoCjUrStk4Id5Gx1h1hJWoURtaAgpRYzWtR4NROR0JWfRIzjU1mTHj%2FffI7awMEjSkC81fNThvJP0i:QRvCd6%2F5c5XutC21KbZ6PG:KxyOuvDQ66CpbxSV91iYOONh0Eg6x8%2Bs00813I7%2BV2Tto%2FCHT5qSt%2BE8gejb52s6diQvzb14%2F2FYqg3GKKwybYnpSpua0b6T5GmH8uKvrGFzn4Xnm6S9icsoyjbUqoWqKc3%2B8W>

Request Method: GET

Status Code: 302 Found

Remote Address: 128.206.56.10:443

Referrer Policy: no-referrer-when-downgrade

Response Headers

Cache-Control: no-store

Connection: close

Content-Length: 0

Content-Type: text/plain; charset=UTF-8

Date: Sat, 04 Nov 2017 20:13:35 GMT

Location: /idp/profile/SAML2/Redirect/SSO?execution=e4s1

Request Headers

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.8

Connection: keep-alive

Cookie: JSESSIONID=DB90C766D301D138769F7DA30C3F7D22

Host: shib-idp.umsystem.edu

7 / 17 requests | 11.5 KB / 11.5 KB...

SAML

Request Response

```
1 <samlp:AuthnRequest
2   AssertionConsumerServiceURL="https://mst.instructure.com/login/saml"
3   ID="a022b969dedd0501e0f61b6c90be8cdf0dd5315e1" IssueInstant="2017-11-
4   04T21:36:26Z"
5   ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Version="2.0"
6   xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
7   <saml:Issuer
8     xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">http://mst.instructure.com/saml2<
9     /saml:Issuer>
10    <samlp:NameIDPolicy AllowCreate="true"
11      Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"
12      xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"></samlp:NameIDPolicy>
13    <samlp:RequestedAuthnContext Comparison="exact"
14      xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
15      <saml:AuthnContextClassRef
16        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">urn:oasis:names:tc:SAML:2.0:ac:c1
17        asses:PasswordProtectedTransport</saml:AuthnContextClassRef>
18      </saml:RequestedAuthnContext>
19    </samlp:AuthnRequest>
```

How Shibboleth Works at S&T – IdP

- > Our session-identifying cookie still matches
 - Prevents masquerading, someone interrupting the transaction in the middle

The screenshot displays the 'Network' tab of a web browser's developer tools. The left sidebar shows a list of network requests, with 'SSO?execution=e4s1' selected. The main pane shows the details for this request:


- General:**
 - Request URL: `https://shib-idp.umsystem.edu/idp/profile/SAML2/Redirect/SSO?execution=e4s1`
 - Request Method: GET
 - Status Code: 200 OK
 - Remote Address: 128.206.56.10:443
 - Referrer Policy: no-referrer-when-downgrade
- Response Headers:**
 - Cache-Control: no-store
 - Connection: close
 - Content-Length: 5625
 - Content-Type: text/html; charset=utf-8
 - Date: Sat, 04 Nov 2017 20:13:35 GMT
- Request Headers:**
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
 - Accept-Encoding: gzip, deflate, br
 - Accept-Language: en-US,en;q=0.8
 - Connection: keep-alive
 - Cookie: JSESSIONID=DB90C766D301D138769F7DA30C3F7D22
 - Host: shib-idp.umsystem.edu
 - Upgrade-Insecure-Requests: 1
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36
- Query String Parameters:**
 - execution: e4s1

At the bottom left, it indicates '7 / 17 requests | 11.5 KB / 11.5 KB...'.

How Shibboleth Works at S&T – IdP

Shibboleth Identity Provider

Secure | <https://shib-idp.umsystem.edu/idp/profile/SAML2/Redirect/SSO?sessionid=A1C886A3D22214F0F2D...>

 **University of Missouri System**
COLUMBIA | KANSAS CITY | ROLLA | ST. LOUIS


Username

Password





Login

[Help](#) [Using a shared computer?](#)

You are currently logging into:
Missouri University Science and Technology - Canvas

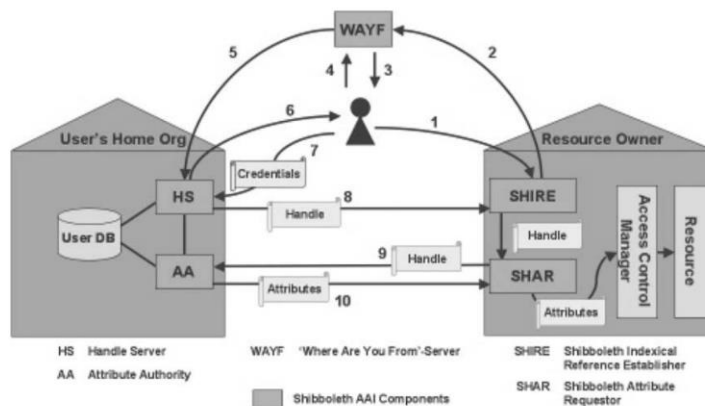
 **canvas**

Missouri University Science and Technology - Canvas

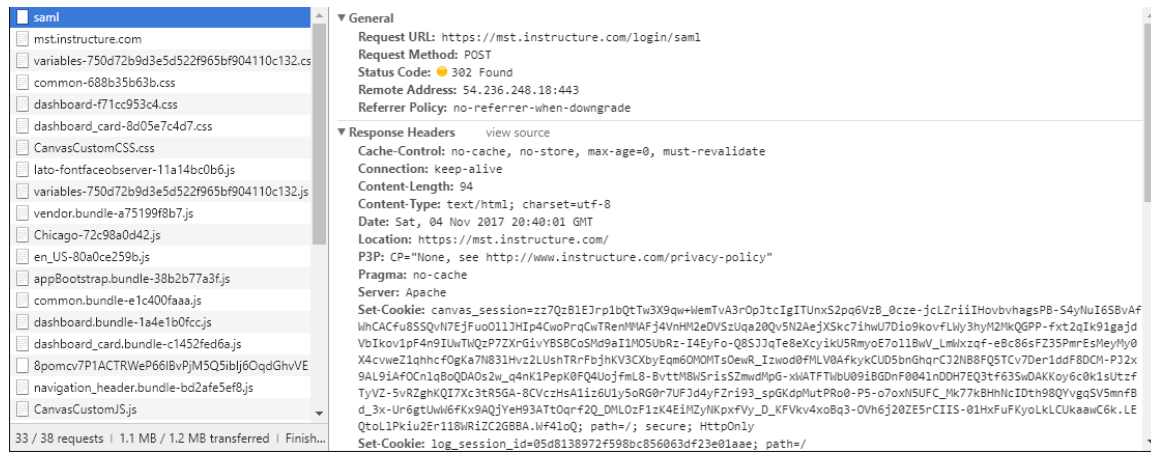
How Shibboleth Works at S&T – Missing Something?

- > No WAYF evident in this interaction.
 - University of Missouri hosts one central active directory for single sign-ons.
 - Technically, everyone comes from the same organization, so users are always directed strait to the “University of Missouri System” login
- > The previous covered steps 1-6, but skipped 3 and 4 for this reason



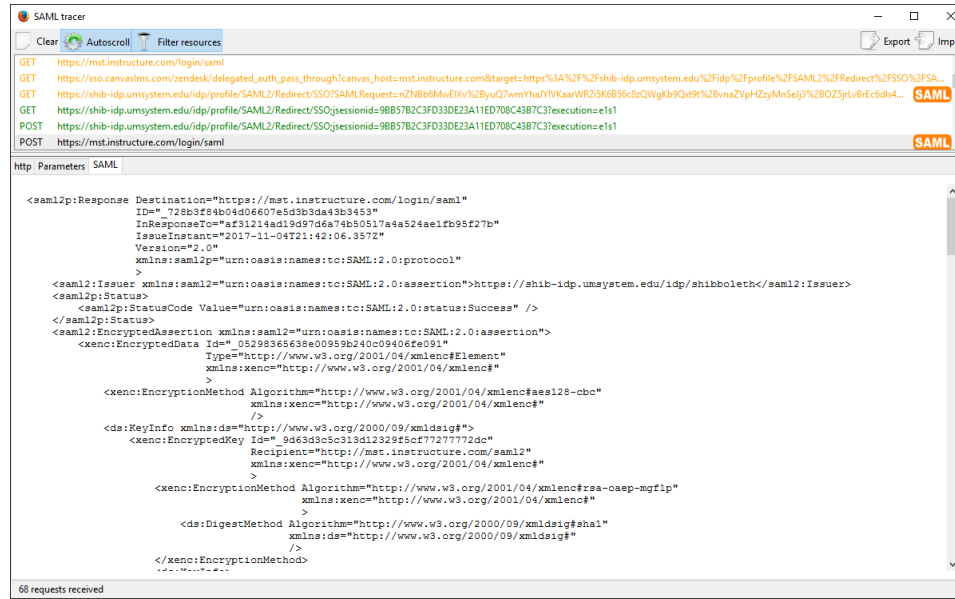
How Shibboleth Works at S&T – ReP

- > After login, we get an SAML response



How Shibboleth Works at S&T – SAML Response

- > SAML Header information
- > Cipher algorithm is 128-bit AES, CBC



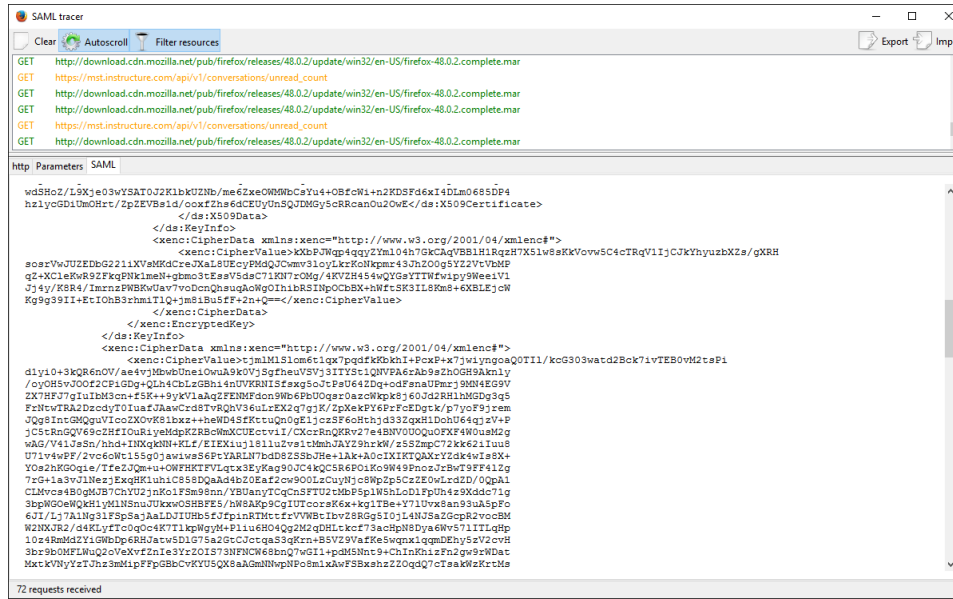
The screenshot shows a web browser window titled "SAML tracer". The address bar displays a series of HTTP requests and responses. The main content area shows the XML structure of a SAML response. The response is a SAML 2.0 Response with a Destination of "https://mat.instructure.com/login/saml". It contains an Issuer, a Status Code of "Success", and an Encrypted Assertion. The Encrypted Assertion is encrypted using a 128-bit AES algorithm in CBC mode. The key is encrypted using a SHA1 digest and a recipient of "https://mat.instructure.com/saml2". The digest is also encrypted using a SHA1 digest and a recipient of "https://mat.instructure.com/saml2".

```
<saml2p:Response Destination="https://mat.instructure.com/login/saml"
ID="728b3f84b04d06607e5d3b3da43b3453"
InResponseTo="af31214ad19d97d6a74b50517a4a524ae1fb95f27b"
IssueInstant="2017-11-04T21:42:06.357Z"
Version="2.0"
xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
>
<saml2:Issuer xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" https://shib-idp.umsystem.edu/idp/shibboleth/>saml2:Issuer</saml2:Issuer>
<saml2p:Status>
<saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" />
</saml2p:Status>
<saml2:EncryptedAssertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
<xenc:EncryptedData Id="052983668e00959b240c09406e0991"
Type="http://www.w3.org/2001/04/xmlenc#Element"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
>
<xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
/>
<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
>
<xenc:EncryptedKey Id="9d63d3d3c313d12325f3c7727772dc"
Recipient="http://mat.instructure.com/saml2"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
>
<xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
/>
</xenc:EncryptionMethod>
</ds:KeyInfo>
</saml2:EncryptedAssertion>
</saml2p:Response>
```

MISSOURI
S&T[illegible]

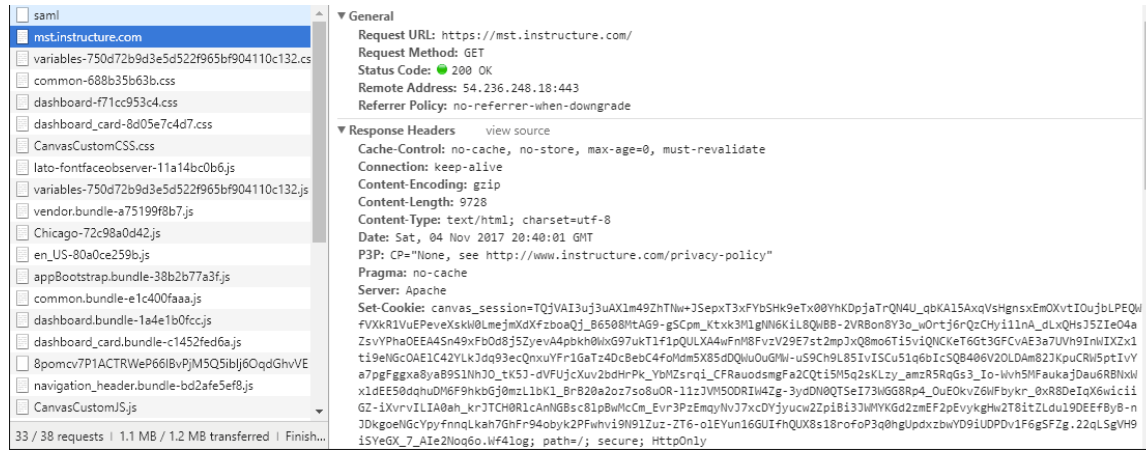
MISSOURI
S&T

- Can't decrypt this without knowing Canvas' private key



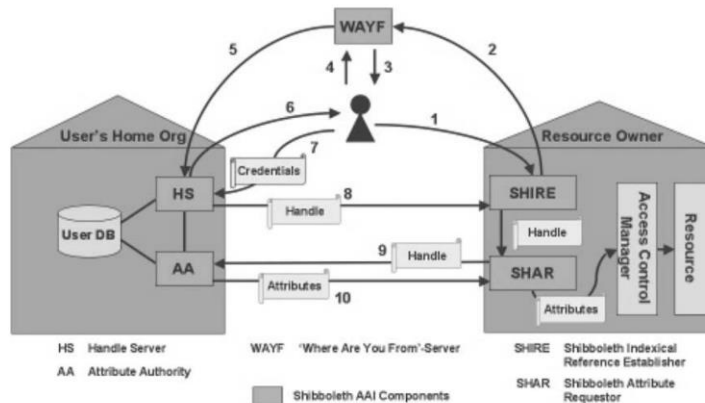
How Shibboleth Works at S&T – ReP

- > SAML response decrypted by ReP
 - Assuming the user had the proper attributes to allow access
 - ReP can now see all it needs to see (but no more) in the SAML response



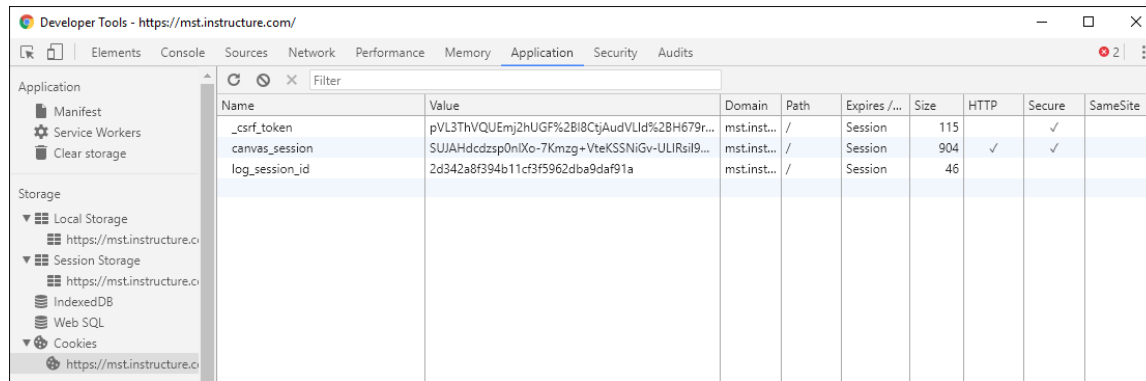
How Shibboleth Works at S&T – Final Notes

- > The previous covered steps 8-10
 - After step 7, everything is opaque to the user
- > Shibboleth is no longer needed for the lifetime of this session at this ReP
- > User interacts directly with Canvas



How Shibboleth Works at S&T – Final Notes

- Two cookies are now set and stay with the user throughout their interactions
 - `canvas_session` – base64 encoded string, probably contains attribute information from Shibboleth
 - `log_session_id` – unique identifier for this session



Moving Forward

(at time of publishing in 2004)

- > Shibboleth targeted at academic/research institutions
- > Author is from UK, participating in discussions to make sure Shibboleth terminology accommodates non-US-based conventions
- > Securing funding for creation of a national Core Middleware Infrastructure, to support UK academic institutions in matters such as this
- > Athens service being modified to interface with Shibboleth and accommodate some of its principles

Shibboleth Today

- > Owned by Internet2, a US-based not-for-profit consortium
- > Shibboleth software is open-source
 - OpenSAML-C++ and OpenSAML-Java libraries available
- > Shibboleth openly lists 84 users of their product
 - From all across the world
 - Includes the University of Missouri
- > Receives commercial support from a number of corporations
- > Version 3.0 released in 2014
 - Incremental releases for version 3.x.x (currently on 3.3.2)
 - University of Missouri is still on version 2.0
 - Most improvements involve adding support for new and expanding technologies

Questions?