ASREN Identity Federation Training

Identity Federation key components

www.geant.org
Content

• What is **Security Assertion Markup Language (SAML)**?
  • SAML V2.0 Deployment Profile for Federation Interoperability
  • SAML Components
• Example of **Authentication flow**
• SAML Metadata
  • Federation Metadata
  • Federation Metadata Structure
• Add Trust to Metadata
  • Certificates usage between IdPs and SPs
  • Certificates usage at the Federation level
• SAML2 IdP and SP Implementations
  • SAML2 IdP and SP Selection criteria
• Introduction to Shibboleth
  • Shibboleth Service Provider
  • Shibboleth Identity Provider
• Attributes exchange
  • Attributes Specification
  • Attribute Schemas
• Recommended Attributes in eduGAIN
• Identifiers Attributes
What is Security Assertion Markup Language (SAML)?

The Security Assertion Markup Language is an **XML-based, open-standard data format for exchanging authentication and authorization data between parties**, in particular, between an Identity Provider and a Service Provider.

**SAML - Security Assertion Markup Language**
- OASIS standard describing the XML messages exchanged
- between Identity Provider (IdP) and Service Provider (SP)
- Its purpose is to enable the **authentication and secure exchange trusted identity information (attributes)** between IdP and SP

**SAML 2.0** is the de facto standard for academic identity federations.

**SAML 2.0 Web Browser Single Sign On Profile**
- Profile that describes how to use SAML in order to achieve SSO
- Main use of SAML within Identity Federations
SAML V2.0 Deployment Profile for Federation Interoperability

This profile specifies behavior and options that deployments of the SAML V2.0 Web Browser SSO profile, and related profiles, are required or permitted to rely on.

SAML2int Interoperable SAML 2.0 Profile

• A deployment profile for SAML2.0 Web Browser Single Sign On profile
• Aims to influence how a SAML entity should be implemented
• Aims to influence how a SAML entity implementation should be configured
SAML Components

SAML Metadata
An XML document describing SAML Entities, both in technical as well as non-technical terms.

Valid SAML Metadata MUST meet the requirements defined in the SAML Metadata Specification

SAML Metadata Consumer
An entity or organization that downloads, processes and uses SAML V2.0 Metadata.

SAML Metadata Producer
An organization that produces and publishes SAML V2.0 Metadata. An XML document describing SAML Entities
SAML Metadata Consumers/Producers

<table>
<thead>
<tr>
<th>At the institution level</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IdP</strong></td>
</tr>
<tr>
<td>• A <strong>SAML authority that authenticates users</strong> against a user repository</td>
</tr>
<tr>
<td>• Retrieves information for the users in the form of attributes</td>
</tr>
<tr>
<td>• Transfers the authentication event along with the attributes to a SAML Service Provider</td>
</tr>
</tbody>
</table>

| **SP**                   |
| • A **SAML consumer** that acts as a middleware in order to protect Web Applications |
| • Consumes SAML messages from the Identity Provider and deduces authentication events and attributes |

| **Discovery Service**    |
| • The Discovery Service service, also known as "Where Are You From (WAYF)" service, **lets the user choose his home institution from a list and then redirects the user to the login page** of the selected institution for authentication. |
## SAML Metadata Consumers/Producers

<table>
<thead>
<tr>
<th>At the NREN level</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Federation Registry</strong></td>
<td>The system component that helps Federation Operators to register and manage entities. It could be used for collecting, processing and republishing federation metadata.</td>
</tr>
<tr>
<td><strong>Central Discovery Service (optional)</strong></td>
<td>A central Discovery Service, operated by Federation Operators to support and help their institutions to deploy services without having to deploy a discovery service at their level.</td>
</tr>
</tbody>
</table>
Example of Authentication flow

Access Service

Choose Home Organisation

Provide credentials

Use Service

Where Are You From?

University A

University B

Hospital C

Select your Home Organization

https://www.resource.ex/

https://www.wayf.ex/

https://www.uni-b.ex/

Authenticate yourself

User name

hmuster

Password

****

Submit

Cancel

OK

Welcome at Resource

Hans Muster's Content:

Your Online Content:

Medical Training 1

Medical Training 2

Live Stream 3

...
SAML Authentication Flow

1. User Agent attempts to access some resource at the Service Provider

2. Shibboleth Authentication Request issued by Service Provider to WAYF Service or Identity Provider

3. If WAYF used, then User Agent selects Identity Provider to which Authentication Request is redirected

4. Identity Provider identifies Principal (methods vary, details not shown)

5. `<saml:Response>` message issued by Identity Provider to Service Provider, and MAY contain SAML attributes

6. Service Provider sends `<saml:AttributeQuery>` to Identity Provider

7. Identity Provider returns `<saml:Assertion>` to Service Provider

8. Based on the Identity Provider's assertion(s) identifying the Principal, the Service Provider either returns the resource or an (HTTP) error
SAML Metadata

• Entities in a federation trust each other, but which entities are in federation?

• Entities in a federation must be known to trust them. Therefore, a standard way to list and describe entities is needed.

• (SAML2) Metadata provides such a standard way. Standardized format to describe entities

• Metadata typically is signed by a trusted third party, in our case the federation operator which should be trusted by all participants in the federation.
Federation Metadata

• The **federation Metadata** provides the technical trust in the federation
  • XML Documents defined by the SAML 2.0 standards
  • Generated by the Federation operators
  • Cryptographically signed by the Federations operators
  • Optionally transported over the internet using SSL
  • Contains technical information on all participating entities
Federation Metadata Structure

• To trust the metadata, it should be **protected** properly.

• No defined order of IdPs and SPs.

• Other entities could be described too. But mostly IdPs and SPs.
Add Trust to Metadata

• **Consumers** of metadata must be sure that the metadata was really created by Federation Operator

• Therefore, **metadata must be secured**

• Two methods to secure metadata:
  
  A. **Recommended:** Add an **XML signature** on metadata and publish public signing key. Metadata can be served via http in this case.
  
  B. Serve plain metadata via a **secure HTTPS URL**. Make web server use a certificate issued by a well-known CA
Certificates usage between IdPs and SPs

• **SSL/TLS** between the user’s browser and the Web server:
  • indicates that the Web Service protects user data and ensures that the user is connected to an authentic site.

• **Self-signed certificate** (and private key) for signing/encrypting, with long lifetime (> 10y)
  • The signing of SAML assertions allows the recipient of a SAML assertion to verify the identity of the issuer and to verify its integrity. This operation is carried out by the two parties involved in the exchange (IDP and SP).
  • **Encryption of SAML assertions** by using the recipient's certificate to encrypt the assertion, the recipient will be able to access the content of the assertion.
Certificates usage between IdPs and SPs
Certificates usage at the Federation level

• Federation operator signing certificate (the metadata file published by the federation operator is signed):
  • eduGAIN collects the metadata of all the participating federations, re-signs and publishes the aggregated metadata for the interfederation so that it can be consumed by all the participating Federations.
  • In order to be able to validate the integrity and authenticity of the Federation’s metadata, the eduGAIN Operations Team needs to receive the certificate with which the Federation signs its locally aggregated metadata.
  • This certificate is also used by IdPs and SPs to ensure that the integrity and authenticity of the Federation’s metadata.
SAML2 IdP and SP Implementations

- **Shibboleth** (most common in academic environment, most comprehensive set of features)
- **SimpleSAMLPHP** (second most common, mostly suitable for PHP applications)
- **Microsoft ADFS** (limited SAML2 support, much handwork to get it running in a federation)
- **pySAML2** (python implementation)
- **mod_mellon** (Apache module, small user base)
- ................
SAML2 IdP and SP Selection criteria

• Can it consume SAML2 metadata?
  • Containing > 3k entities
  • More than 22MB in size
  • Can metadata be refreshed automatically

• Does it support the Web SSO profile (saml2int.org)?
  • E.g. can it process signed and encrypted SAML assertions from IdPs with self-signed X.509 certificates?

• Is it a secure, well adopted implementation backed by a strong community or vendor?
Attributes exchange

• When a user attempts to access a Service Provider-protected site, it usually asks the user’s Identity Provider to provide one or more specific identity attributes.

• Example:
  
  ![Diagram showing the flow of identity attributes]
  
  - **LDAP**
  - **IdP**
  - **SAML2**
  - **SP**
  - **Application**

  Attribute « mail »
  Value: john.doe@univ.ac.cy

  IdP allows or not the release of the attribute to the SP

  urn:oid:0.9.2342.19200300.100.1.3
Introduction to Shibboleth

• The Origin
  • Internet2 in the US launched the open source project in 2000

• The name
  • Word Shibboleth was used to identify members of a group

• The standard
  • Based on Security Assertion Markup Language (SAML)

• The Consortium
  • The new home for Shibboleth development
  • Collect financial contributions from deployers worldwide

• The Shibboleth software is the most widely used in the research and education environment

• Website: https://shibboleth.net/
Shibboleth Service Provider

- Runs on: Linux, Solaris, Windows, Mac OS X, FreeBSD, ...
- **Protects web applications**
- shibd processes attributes
- **Can authorize users with**
  - Apache directives
  - Shibboleth XML
  - Access rule
- **Provides attributes to applications** via web server environment variables or headers
Shibboleth Identity Provider

- Runs on: Linux, Solaris, Windows, Mac OS X, FreeBSD, ...
- Authentication
  - Attribute Resolution
    - **pulling in data** from external systems (e.g., LDAP directories and relational databases)
    - **Creating attributes** from the pulled in data
    - **Associating protocol-specific encoders with the created attributes.**
  - Attribute Filter
    - Determines what information are allowed to be sent to a requester.
  - Relying party
    - Define SAML profiles for SPs
Authentication

- Based on Spring Web Flow

- Login flows
  - **Password**
  - RemoteUser
  - RemoteUserInternal
  - X509
  - X509Internal
  - SPNEGO / Kerberos
  - IPAddress
  - External
  - Duo
  - Multi-Factor
  - Function
  - SAML

*Most popular*

**Where to specify which authentication flow to use?**

```
/opt/shibboleth-idp/conf/idp.properties:
« idp.authn.flows = Password »
```

**Where can I find available flows?**

```
/opt/shibboleth-idp/auth/....
```
Attribute Resolver

• Sample files provided by default
  • /opt/shibboleth-idp/conf/attribute-resolver.xml
  • <shibboleth-idp-src-path>/conf/examples/attribute-resolver-ldap.xml

• Where to choose the right file to take into account ?
  • /opt/shibboleth-idp/conf/services.xml

• It contains :
  • DataConnector
  • AttributeDefinition
DataConnectors

Defines connections to sources of data which provide input to attribute definitions.

### DataConnector Plugin Types

<table>
<thead>
<tr>
<th>Plugin Type</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>+++</td>
</tr>
<tr>
<td>ScriptedDataConnector</td>
<td>+++</td>
</tr>
<tr>
<td>ComputedId</td>
<td>++</td>
</tr>
<tr>
<td>StoredId</td>
<td>+++</td>
</tr>
<tr>
<td>Pairwiseld</td>
<td>++</td>
</tr>
<tr>
<td>RelationalDatabase</td>
<td>++</td>
</tr>
<tr>
<td>LDAPDirectory</td>
<td>+++</td>
</tr>
<tr>
<td>HTTP</td>
<td></td>
</tr>
<tr>
<td>Subject</td>
<td></td>
</tr>
<tr>
<td>StorageService(^{4.1})</td>
<td>++</td>
</tr>
</tbody>
</table>
Data Connector (example: LDAPDirectory)

```xml
<DataConnector id="myLDAP" xsi:type="LDAPDirectory"
    ldapURL="%{ldap.attribute.resolver.LDAP.ldapURL}"
    baseDN="%{ldap.attribute.resolver.LDAP.baseDN}"
    principal="%{ldap.attribute.resolver.LDAP.bindDN}"
    principalCredential="%{ldap.attribute.resolver.LDAP.bindDNCredential}"
    useStartTLS="%{ldap.attribute.resolver.LDAP.useStartTLS:true}"%
    connectTimeout="%{ldap.attribute.resolver.LDAP.connectTimeout}"%
    trustFile="%{ldap.attribute.resolver.LDAP.trustCertificates}"%
    responseTimeout="%{ldap.attribute.resolver.LDAP.responseTimeout}"%
    connectionStrategy="%{ldap.attribute.resolver.LDAP.connectionStrategy}"%
    noResultsIsError="true"
    multipleResultsIsError="true"
    excludeResolutionPhases="cli4n/attribute"
    exportAttributes="mail displayName sn givenName departmentNumber employeeNumber eduPersonEntitlement eduPersonAssurance">
    <![CDATA[
        "%{ldap.attribute.resolver.LDAP.searchFilter}"
    ]]>%
</FilterTemplate>
</ConnectionPool>
```

Data Connector (example: Static)

<!--  Data Connectors  -->

<DataConnector id="staticAttributes" xsi:type="Static">
  <Attribute id="affiliation">
    <Value>member</Value>
  </Attribute>
</DataConnector>
### Attribute Resolver

<table>
<thead>
<tr>
<th>Attribute Definition</th>
<th>++</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>+++</td>
</tr>
<tr>
<td>PrincipalName</td>
<td>+++</td>
</tr>
<tr>
<td>Scoped</td>
<td>+++</td>
</tr>
<tr>
<td>Prescoped</td>
<td>+++</td>
</tr>
<tr>
<td>RegexSplit</td>
<td>+++</td>
</tr>
<tr>
<td>ScriptedAttribute</td>
<td>+++</td>
</tr>
<tr>
<td>Mapped</td>
<td>+++</td>
</tr>
<tr>
<td>Template</td>
<td></td>
</tr>
<tr>
<td>SubjectDerived</td>
<td></td>
</tr>
<tr>
<td>ContextDerived</td>
<td></td>
</tr>
<tr>
<td>Decrypted$^{4.1}$</td>
<td></td>
</tr>
</tbody>
</table>

Which Attribute Definition to choose?
Attribute Resolver (examples)

Simple attribute definition       john.doe

<AttributeDefinition xsi:type="Simple" id="uid">
    <InputDataConnector ref="myLDAP" attributeNames="sAMAccountName" />
</AttributeDefinition>

Scoped attribute definition       member@example.org

<AttributeDefinition scope="%{idp.scope}" xsi:type="Scoped" id="eduPersonScopedAffiliation">
    <InputDataConnector ref="myLDAP" attributeNames="eduPersonAffiliation" />
</AttributeDefinition>
Attributes Specification

• **Attribute Specification** is crucial for the data exchange within a federation or eduGAIN, it provides the common basis on which two communicating entities are able to share information they know to interpret identically.

• **Federation participants must support a defined set of user attributes** and requires or recommends the use or release of user attributes to enable interoperability, good user experience, and to help protect personal privacy. A supported attribute might or might not be made available for use by an Identity Provider.
Attribute Schemas

MyEduPerson

https://wiki.refeds.org/display/STAN/SCHAC+Releases
https://wiki.refeds.org/display/STAN/eduPerson+2020-01
### Recommended Attributes in eduGAIN: 1/2

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>commonName</td>
<td>Name and Surname of the user.</td>
</tr>
<tr>
<td><strong>OID</strong>: 2.5.4.3</td>
<td><strong>Example</strong>: John Doe</td>
</tr>
<tr>
<td>displayName</td>
<td>Name and Surname of the user.</td>
</tr>
<tr>
<td><strong>OID</strong>: 2.16.840.1.113730.3.1.241</td>
<td><strong>Example</strong>: John Doe</td>
</tr>
<tr>
<td>mail</td>
<td>Holds Internet mail address</td>
</tr>
<tr>
<td><strong>OID</strong>: 0.9.2342.19200300.100.1.3</td>
<td><strong>Example</strong>: <a href="mailto:john.doe@example.org">john.doe@example.org</a></td>
</tr>
<tr>
<td>schacHomeOrganization</td>
<td>Specifies a person’s home organization using the domain name of the organization.</td>
</tr>
<tr>
<td><strong>OID</strong>: 1.3.6.1.4.1.25178.1.2.9</td>
<td><strong>Example</strong>: example.org</td>
</tr>
<tr>
<td>schacHomeOrganizationType</td>
<td>Type of a Home Organization</td>
</tr>
<tr>
<td><strong>OID</strong>: 1.3.6.1.4.1.25178.1.2.10</td>
<td><strong>Example</strong>: urn:schac:homeOrganizationType:int:university</td>
</tr>
</tbody>
</table>
Recommended Attributes in eduGAIN: 2/2

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eduPersonAffiliation</td>
<td>Specifies the person's relationship(s) to the institution in broad categories. Example: such as student, faculty, staff, alum, etc.</td>
</tr>
<tr>
<td>eduPersonPrincipalName (ePPN)</td>
<td>Unique, persistent identifier of the user. Example: <a href="mailto:jdoe@example.org">jdoe@example.org</a></td>
</tr>
<tr>
<td>eduPersonScopedAffiliation</td>
<td>Specifies the person's affiliation within a particular security domain. Example: <a href="mailto:staff@example.org">staff@example.org</a>; <a href="mailto:member@example.org">member@example.org</a></td>
</tr>
<tr>
<td>eduPersonTargetedID/persistentID (ePTID)</td>
<td>Unique, persistent, opaque and targeted identifier of the user. (Serialized) Example: <a href="https://idp.example.org/idp/shibboleth!http://sp.example.org!yrdfefohZY+cdGvqu/Dubc=">https://idp.example.org/idp/shibboleth!http://sp.example.org!yrdfefohZY+cdGvqu/Dubc=</a></td>
</tr>
</tbody>
</table>
Identifiers Attributes

• Web services often need to **re-identify users** in order to present them their profile, and should receive only the absolutely needed information about the users (data minimization principle).

• Properties of identifiers:
  • **Uniqueness**: as the name implies an identifier should identify a user without doubt, no two users should have the same identifier.
  • **Reassignability**: user identifiers can be reassigned to another user after the first user with this identifier has left the organization. Reassigned identifiers can cause access control and traceability problems.
  • **Opacity**: an identifier that gives no clue (i.e. it only contains random data and no names) about the identity of the user is called opaque.
  • **Persistency**: an identifier that stays identical over time is called persistent.
  • **Targetedness**: an identifier that is intended to be used for a single service only and is specific to that service is called targeted.
  • **Transientness**: a transient identifier stays the same for a login session, but changes when the user logs in again.
### Overview of identifier attributes and their characteristics

<table>
<thead>
<tr>
<th>Property/Identifier</th>
<th>ePPN</th>
<th>ePTID</th>
<th>mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Non-reassignable</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opaque</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Persistent</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Targeted</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transient</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Availability in eduGAIN</td>
<td>Very good</td>
<td>Good</td>
<td>Very good</td>
</tr>
</tbody>
</table>
Attribute Filter

• The Shibboleth IdP allows filtering outgoing user attributes
  • To limit the distribution of nominative attributes to services
• /opt/shibboleth-idp/conf/attribute-filter.xml
  • Defines rules for each SP
  • Defines rules for each attribute
• By default, this file is poor in filtering rules
  • no attribute sent

Side effect of attribute filtering at IdP level ➔ If the IdP is too restrictive, users will not have access to services (SP)
Attribute Filter (example)

<!-- Release an additional attribute to an SP. -->
<AttributeFilterPolicy id="example1">
  <PolicyRequirementRule xsi:type="Requester" value="https://sp.example.org" />
  <AttributeRule attributeID="uid" permitAny="true" />
</AttributeFilterPolicy>

<!-- Release eduPersonScopedAffiliation to two specific SPs. -->
<AttributeFilterPolicy id="example2">
  <PolicyRequirementRule xsi:type="OR">
    <Rule xsi:type="Requester" value="https://sp.example.org" />
    <Rule xsi:type="Requester" value="https://another.example.org/shibboleth" />
  </PolicyRequirementRule>
  <AttributeRule attributeID="eduPersonScopedAffiliation" permitAny="true" />
</AttributeFilterPolicy>
But how IdPs knows what attributes are requested by each service?

From the attribute filter, IdP administrators are defining which attributes to send to each service. eduGAIN contains more than 3000 services, might be time consuming to define attributes for each one.

- Use of Dynamic Filter configuration:
  - Requested attributes could be shared in the metadata (SP or federation metadata)
  - IdPs can be configured to release automatically these attributes if available in metadata
Thank you

Any questions?

www.geant.org