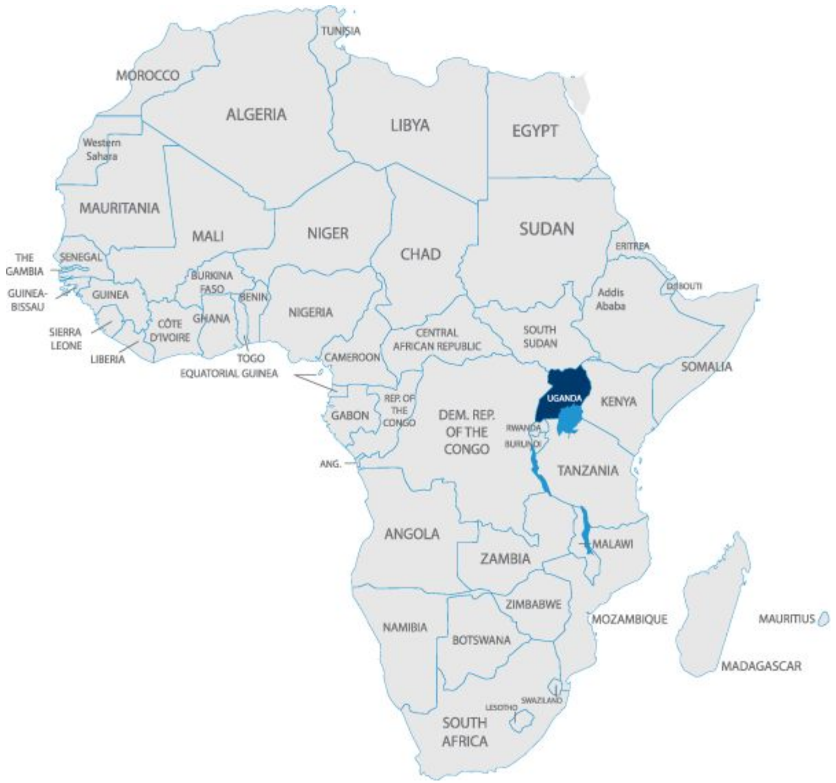
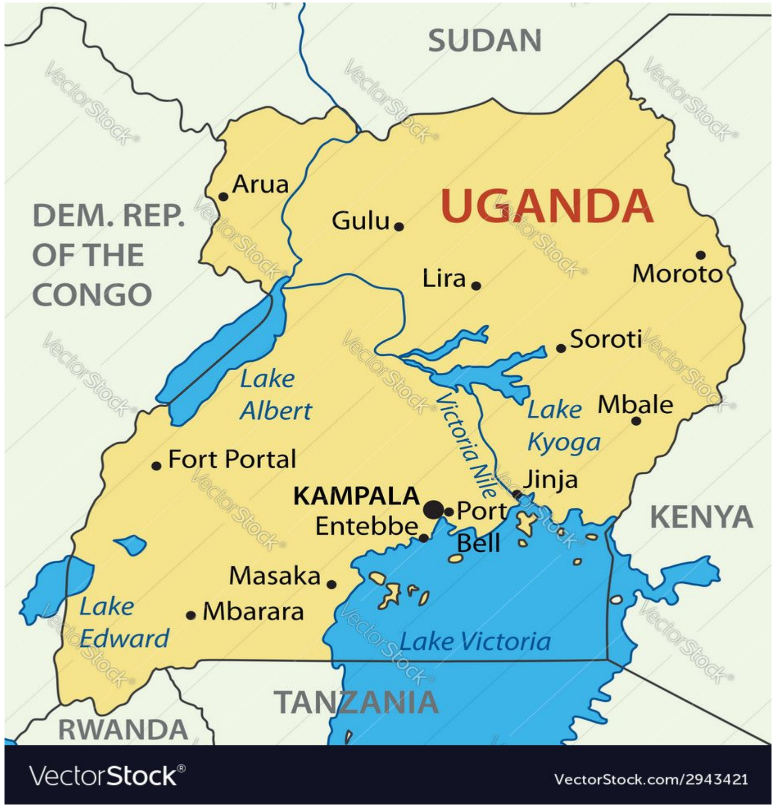
SAML & SAML Federations

Mwotil Alex

RENU Identity Federation (RIF)

Acknowledgment: Scott Koranda


MOROCCO
ALGERIA
TUNISIA
LIBYA
EGYPT
Western Sahara
MAURITANIA
MALI
NIGER
CHAD
SUDAN
ERITREA
THE GAMBIA
SENEGAL
GUINEA-BISSAU
GUINEA
BURKINA FASO
BENIN
NIGERIA
Addis Ababa
DJIBOUTI
SIERRA LEONE
CÔTE D'IVOIRE
GHANA
TOGO
LIBERIA
CENTRAL AFRICAN REPUBLIC
SOUTH SUDAN
CAMEROON
EQUATORIAL GUINEA
SOMALIA
UGANDA
KENYA
GABON
REP. OF THE CONGO
DEM. REP. OF THE CONGO
RWANDA
BURUNDI
ANG.
TANZANIA
ANGOLA
ZAMBIA
MALAWI
ZIMBABWE
MOZAMBIQUE
MAURITIUS
NAMIBIA
BOTSWANA
MADAGASCAR
SWAZILAND
LESOTHO
SOUTH AFRICA

SUDAN
DEM. REP. OF THE CONGO
UGANDA
Arua
Gulu
Lira
Moroto
Soroti
Lake Albert
Victoria Nile
Lake Kyoga
Mbale
Fort Portal
Jinja
KAMPALA
Port Bell
Entebbe
KENYA
Masaka
Mbarara
Lake Edward
Lake Victoria
TANZANIA
RWANDA
VectorStock®
VectorStock.com/2943421

- AKA *'the Pearl of Africa'*
- A rolex is not a watch
- Over 52 tribes
- Most treasured bird - the crane

Outline

- Introduction to XML
- Introduction to SAML
- How SAML Works
- Elements of SAML
- Transport Protocol (Bindings)
- SAML Messages
- SAML Metadata
- Security
- Attributes

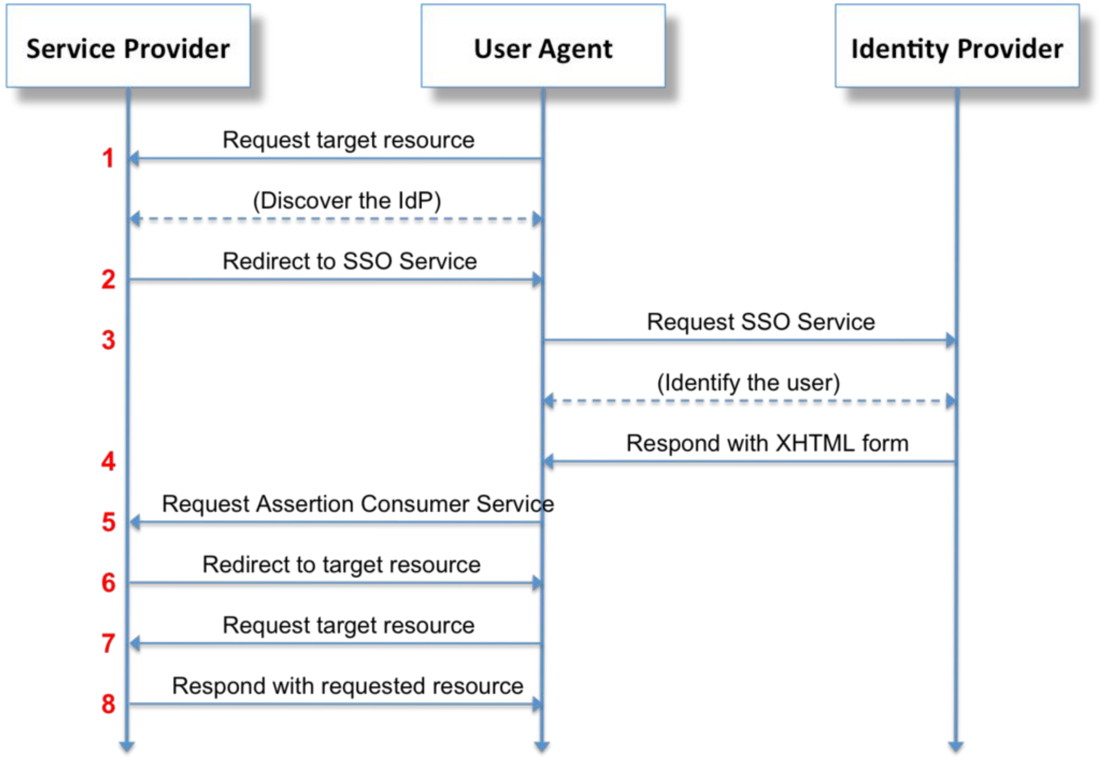
SAML

- SAML - Security Assertion Markup Language
- Security framework and open standard defined by OASIS
  - series of technical documents and XML schemas
- Focus on SAML web browser single sign-on profile (SAML WebSSO)
- Focus still further on SAML2 Interoperability Deployment Profile V1.0
  - SAML2int
  - Designed by and for higher education and research to improve interoperability
  - https://kantarainitiative.github.io/SAMLprofiles/saml2int.html
  - Version 2 still a draft

SAML - Parts

- A SAML transaction is made up of three parts:
  - An authentication (Authn) request
  - An authentication response
  - Assertions about subject of authentication
    - Expressed as attributes
    - User identification
- Assertions may provide information required for authorization

SAML Web Browser SSO



SAML Web Browser SSO

1. Request the target resource at the SP via an HTTP user agent
2. Redirect to the SSO Service at the IdP - SP determines the preferred identity provider
3. Request the SSO Service at the IdP - User agent issues a GET request to the SSO service at the URL
4. Respond with an XHTML form - SSO service validates the request and responds with a document containing an XHTML form
5. Request the Assertion Consumer Service at the SP
6. Redirect to the target resource
7. Request the target resource at the SP again

SAML - Message Transport

There are many ways to transport a SAML message (request or response)

The different ways are referred to as **protocol bindings**

- **HTTP-Redirect:** SAML message passed by redirecting web browser to perform a GET from a URL with the SAML message passed in the query string
- **HTTP-POST:** SAML message passed by delivering it to the web browser and instructing the web browser to push it using HTTP POST (like a web form)

SAML - Message Transport

- HTTP-Redirect and HTTP-POST use web browser to pass SAML message
- Also called "front channel" bindings since the browser is the transport agent
- "Back channel" bindings also exist
  - Direct communication between service provider (SP) and identity provider (IdP)
  - HTTP Artifact binding
  - Much less common in higher education and research

Focus on front channel bindings

SAML - Messages

- Messages are structured as XML - No need to understand this structure
- A couple of useful debugging tools to view SAML messages
  - SAML tracer Add-on for FireFox
  - SAML DevTools extension for Chrome
  - Other tools useable but involve more work
    - LiveHTTPHeaders
    - Safari Web Inspector
    - Fiddler Often combined with https://www.samltool.com/

SAML Messages - AuthN Request

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
        AssertionConsumerServiceURL="https://meetings.renu.ac.ug/Shibboleth.sso/SAML2/POST"
        Destination="https://idp.renu.ac.ug/idp/profile/SAML2/Redirect/SSO"
        ID="_3331eb4b55e882b75ae540974b34981d"
        IssueInstant="2019-02-26T07:19:21Z"
        ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
        Version="2.0"
        >
<saml:Issuer
xmlns:saml="urn:oasis:names:tcx:SAML:2.0:assertion">https://meetings.renu.ac.ug/shibboleth-sp2</sa
ml:Issuer>
<samlp:NameIDPolicy AllowCreate="1"/>
</samlp:AuthnRequest>
```

- SP Issuer SAML entityID
- Every SP and IdP has unique entityID
- Best practice is URL syntax
- Older practice is URN

SAML Messages - AuthN Request

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
        AssertionConsumerServiceURL="https://meetings.renu.ac.ug/Shibboleth.sso/SAML2/POST"
        Destination="https://idp.renu.ac.ug/idp/profile/SAML2/Redirect/SSO"
        ID="_3331eb4b55e882b75ae540974b34981d"
        IssueInstant="2019-02-26T07:19:21Z"
        ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
        Version="2.0"
        >
<saml:Issuer
xmlns:saml="urn:oasis:names:tcx:SAML:2.0:assertion">https://meetings.renu.ac.ug/shibboleth-sp2</sa
ml:Issuer>
<samlp:NameIDPolicy AllowCreate="1"/>
</samlp:AuthnRequest>
```

- Timestamp
- Prevents replay attacks
- May tolerate clock skew (3-5 mins)

SAML Messages - AuthN Request

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
        AssertionConsumerServiceURL="https://meetings.renu.ac.ug/Shibboleth.sso/SAML2/POST"
        Destination="https://idp.renu.ac.ug/idp/profile/SAML2/Redirect/SSO"
        ID="_3331eb4b55e882b75ae540974b34981d"
        IssueInstant="2019-02-26T07:19:21Z"
        ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
        Version="2.0"
        >
<saml:Issuer
xmlns:saml="urn:oasis:names:tcx:SAML:2.0:assertion">https://meetings.renu.ac.ug/shibboleth-sp2</sa
ml:Issuer>
<samlp:NameIDPolicy AllowCreate="1"/>
</samlp:AuthnRequest>
```

- URL at the destination (IdP) to consume the request

SAML Messages - AuthN Request

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
        AssertionConsumerServiceURL="https://meetings.renu.ac.ug/Shibboleth.sso/SAML2/POST"
        Destination="https://idp.renu.ac.ug/idp/profile/SAML2/Redirect/SSO"
        ID="_3331eb4b55e882b75ae540974b34981d"
        IssueInstant="2019-02-26T07:19:21Z"
        ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
        Version="2.0"
        >
<saml:Issuer
xmlns:saml="urn:oasis:names:tcx:SAML:2.0:assertion">https://meetings.renu.ac.ug/shibboleth-sp2</sa
ml:Issuer>
<samlp:NameIDPolicy AllowCreate="1"/>
</samlp:AuthnRequest>
```

- SP URL expecting to consume the response

SAML Messages - AuthN Request

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
        AssertionConsumerServiceURL="https://meetings.renu.ac.ug/Shibboleth.sso/SAML2/POST"
        Destination="https://idp.renu.ac.ug/idp/profile/SAML2/Redirect/SSO"
        ID="_3331eb4b55e882b75ae540974b34981d"
        IssueInstant="2019-02-26T07:19:21Z"
        ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
        Version="2.0"
        >
<saml:Issuer
xmlns:saml="urn:oasis:names:tcx:SAML:2.0:assertion">https://meetings.renu.ac.ug/shibboleth-sp2</sa
ml:Issuer>
<samlp:NameIDPolicy AllowCreate="1"/>
</samlp:AuthnRequest>
```

- Protocol binding the SP expects the IdP to use for sending the response

SAML - Response

- IdP uses HTTP-POST binding to send response to SP
  - Base64 encoded XML payload returned to browser
  - **Browser does the POST**
- Most IdPs include Javascript to automate the POST
  - Turn off Javascript and you will see a button to click to force the POST
- More complicated (and therefore longer), because it contains both assertions about the subject and a cryptographic signature

SAML - Response

- Response is usually digitally signed (XML digital signature)
    - SP can verify and trust the response
    - Prevent tampering
    - Response payload may also be encrypted (XML encryption)
    - Encrypted using the SPs SAML key
    - Hides details from user from snooping browsers
    - TLS transport not usually required but usually used
- Includes an assertion about the authentication event
    - Assertion may be encrypted if Response is not

```
<saml2p:Response Destination="https://meetings.renu.ac.ug/Shibboleth.sso/SAML2/POST"
            ID="_bb2aeedfe5c1aac38b4756866504b9c8"
            InResponseTo="_3331eb4b55e882b75ae540974b34981d"
            IssueInstant="2019-02-26T07:19:36.524Z"
            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://idp.renu.ac.ug/shibboleth</saml2:Issuer
>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"> SNIP </ds:Signature>
   <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"> SNIP
</saml2:EncryptedAssertion>
</saml2p:Response>
```

Response to AuthN request

- Authentication was successful

Assertion information and attributes (encrypted)

Encoding SAML Messages

- XML Messages are:
  - Compressed using a DEFLATE mechanism
  - Base64 encoded
  - URL encoded
- Messages can then be sent as well-known parameters in a GET or POST request
  - SAMLRequest for a SAML request, SAMLResponse for a SAML response
  - Signature for the xmldsig signature
  - SigAlg for information about the signature algorithm
  - RelayState as an opaque internal state variable used by the provider.

Encoding SAML Messages - Example

```
GET
https://idp.renu.ac.ug/idp/profile/SAML2/Redirect/SSO?SAMLRequest=fZJLb4MwEIT
%2FCvIdzLNJrIBEk0MjpQ0KtIdeKgPbYAls6jV9%2FPuSkLbpJUfLOzM7n3aJvGt7lg6mkXt4G
wCN9dm1EtnpIyaDlkxxFMgk7wCZqVie3m%2BZ77is18qoSrXEShFBG6HkSkkcOtA56HdRwe
N%2BG5PGmB4ZpR2AEfKAjgY5OLxyhgPNG1GWqgXTOIiKHq19mu3ygljrcRch%2BdH1z0PU
%2FaV8fNJxi1fRwlm7h1poqAzN8x2xNuuYvARB4EEZllEE87lfziIOUeguZmEZhIu5V49jiANsJ
BouTUx811vYrm%2F7N4U7Y96C%2Bd4zsbJz2Vsh67HFdTLlNITsrigye%2BrzBBpPXcYBkiyPf
NkpWF8Qv27LfzCT5BpU%2FIVqY%2B8v6UXUlNuzh9F7s85UK6ovK21b9bHSwA3ExCM0mST
%2FzyL5Bg%3D%3D&RelayState=ss%3Amem%3A5559cee9216ccfa1f08e5a5d503593c3264
9acbbbdf732557b558e7115136199 HTTP/1.1
```

SAML Metadata

- Why should an IdP and SP interoperate?
  - Why should an IdP accept an authentication request from an SP?
  - Why should an IdP authenticate a user and then assert details about that event and identity information to the SP?
  - Why should an SP trust an assertion about a user sent to it from an IdP?

SAML Metadata

- IdP maintains a "list" of trusted relying parties (SPs)
  - Only accept authentication requests from trusted SPs
  - Only send assertions to URLs for SPs that it trusts
  - Should sign/encrypt assertion/response so only the trusted SP can decrypt and consume
- SP maintains a "list" of trusted relying parties (IdPs)
  - Only send authentication requests to trusted IdPs
  - Only send authentication requests to URLs for IdPs that it trusts
  - Only accepts signed/encrypted assertion/response from IdPs that it trusts

SAML Metadata

- SAML Metadata is used to establish Trust
- XML description of the SAML entity
  - entityID
  - SAML role (IdP or SP)
  - URL endpoints for consuming SAML messages
  - signing/encryption public key material
  - organization information including contacts
- SAML metadata, just as SAML messages, is XML
- Most federation operators provide documentation and even templates - https://safire.ac.za/wp-content/uploads/2016/12/metadata.xml

SAML Metadata - Generation & Edits

- SAML software generates the metadata automatically
  - Normally contains extraneous data not relevant to SAML2int
- Metadata (not signed) should be edited to contain only relevant elements
  - Done using a text editor
- Federations provide normalization and validation services of the metadata
- The federation metadata is signed using xmldsig
  - To ensure integrity of the data

```
<EntityDescriptor entityID="https://iziko.safire.ac.za/" >
 <ds:Signature>SNIP</ds:Signature>
 <Extensions><mdrpi:PublicationInfo creationInstant="2019-02-26T08:00:01Z" publisher="https://safire.ac.za"/>
  <mdrpi:RegistrationInfo registrationAuthority="https://safire.ac.za">
   <mdrpi:RegistrationPolicy xml:lang="en">https://safire.ac.za/safire/policy/</mdrpi:RegistrationPolicy>
  </mdrpi:RegistrationInfo>
 </Extensions>
 <IDPSSODescriptor protocolSupportEnumeration=.....>
  <Extensions>
   <mdui:UIInfo>
    <mdui:DisplayName xml:lang="en">SAFIRE - South African Identity Federation</mdui:DisplayName>
    <mdui:Description xml:lang="en">SAFIRE - South African Identity Federation</mdui:Description>
    <mdui:InformationURL xml:lang="en">......</mdui:InformationURL>
    <mdui:PrivacyStatementURL xml:lang="en">........</mdui:PrivacyStatementURL>
    <mdui:Logo width="16" height="16">....../mdui:Logo>
   </mdui:UIInfo>
   <mdui:DiscoHints>
    <mdui:GeolocationHint>.....</mdui:GeolocationHint>
   </mdui:DiscoHints>
   <KeyDescriptor>......</KeyDescriptor>
                                        <SingleLogoutService          Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://iziko.safire.ac.za/saml2/idp/SingleLogoutService.php"/>
  <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</NameIDFormat>
  <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>
                                        <SingleSignOnService          Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://iziko.safire.ac.za/saml2/idp/SSOService.php"/>
 </IDPSSODescriptor>

 <Organization>SNIP</Organization>
 <ContactPerson>SNIP</ContactPerson>
</EntityDescriptor>
```

SAML Metadata - Structure

- Entities have unique identifiers - EntityID, structured in the form of a URL
  - URL host part should be the server FQDN
- RoleDescriptors - Describes that the entity does
  - IDPSSODescriptor or SPSSODescriptor
- Scope
  - IDPSSODescriptor may contain scope elements in realm format
- MDUI
  - Service description and logos
- Entity Categories
  - Group and categorise different entities that share common criteria

SAML Metadata - Security

- SAML digital signing and/or encryption rely on public key cryptography
    - Public key encoded as a certificate is transfered
- Keys may be pre-shared (included in the metadata) or using a Public Key Infrastructure (PKI)
    - Where pre-shared, the explicit trust model applies - IdP/SP explicitly trusts the keys in the
- SAML certificates need not be signed by a certificate authority
    - R&E prefers self-signed certificates
- Public services of the entities requires PKI and should be signed by a certificate authority

SAML Attributes

- Attributes contain information about the users
- Metadata defines the attributes that the service provider requires and where they should be sent to by the identity provider
- Attributes are named using OIDs represented in the URN namespace
- OIDs defined in various LDAP schemas are also used in SAML

SAML Attributes - SAFIRE

- **Required**
  - eduPersonPrincipalName
  - givenName
  - mail
  - sn
- **Recommended**
  - displayName
  - eduPersonAffiliation
  - eduPersonScopedAffiliation