Federated Identity and Access Management

Hands-on with SAML

Here is an overview of the SAML protocol:

To trace the SAML protocol, open Firefox in your VM. From the Firefox menu, select "SAML tracer".
This will open a "SAML tracer" window. Switch back to the primary Firefox window and open https://wiki.refeds.org. That will open the REFEDS Wiki home page. Click "Log in" on that page.

Select your identity provider and log in. You can try using your university's identity provider or a social network (like Facebook or Google). You can also sign up for a guest account at the National Center for Supercomputing Applications (https://go.ncsa.illinois.edu/idp-guest) or United ID (https://unitedid.org/).

Then switch back to the "SAML tracer" window. Deselect "Autoscroll". Look for a "SAML" icon on the right to indicate the SAML messages. First, find a SAML Authentication Request (AuthnRequest):

Use the SAML tab to view the SAML message.

What is the Issuer? ______________________________
What is the Destination? ______________________________

Do you see a second AuthnRequest with a different Issuer/Destination? If yes:
What is the Issuer? ______________________________

What is the Destination? ______________________________

Next, find a SAML IdP Response (saml2p:Response):

![Screenshot of a SAML response](image)

Does the response contain a saml:Assertion or saml2:EncryptedAssertion?

Look for a response containing an EncryptedAssertion:

![Screenshot of an EncryptedAssertion](image)
What is the Issuer of the EncryptedAssertion, if you found one?

Next, look for a response containing a cleartext Assertion:

What is the Issuer of the cleartext Assertion, if you found one?

What Attribute Names and Values do you see?
Finally, Log Out from https://wiki.refeds.org/:

Hands-on with SAML: Advanced

If you have extra time, you can try using different identity providers to Log In to https://wiki.refeds.org/. How do the SAML messages differ?
Hands-on with OpenID Connect

This OpenID Connect section uses the Google OAuth Playground, which requires a Google Account. If you don’t already have a Google Account, you'll need to create one at https://accounts.google.com/SignUp before continuing with this section.

Next, visit the Google OAuth Playground at https://developers.google.com/oauthplayground/. In the text box that says "Input your own scopes", enter "openid email profile" to request 3 standard OpenID Connect scopes:

Then click "Authorized APIs". You will be prompted to log in to your Google account. After logging in, you will receive the OAuth Authorization Code. Next, click "Exchange authorization code for tokens" to obtain a Refresh Token and Access Token. If the "Request / Response" does not appear on the right side, increase the width of your window until it appears.
The response should look similar to the following (with longer token values):

```json
{
    "access_token": "ya29.GlvfBB7urUfLM8nhBULT99ATPe3",
    "token_type": "Bearer",
    "expires_in": 3600,
    "refresh_token": "1/FuZ8UX8dRKe9MNvBEMEY1LqXa4Af3QnfAagodoag",
    "id_token": "eyJhbG0qvEsSGusR6PYbaug"
}
```

Next, open a command-line window and save the access_token value to a shell variable. We'll use it later.

```bash
export ACCESS_TOKEN="ya29.GlvfBB7urUfLM8nhBULT99ATPe3"
```

Next, open [https://jwt.io/](https://jwt.io/) and paste the id_token value into the box under the "ENCODED - PASTE TOKEN HERE" message, and you should see the token decoded on the right side:
Lastly, use the Access Token to obtain user information on the command line. In the command-line window you opened previously, use the ACCESS_TOKEN environment variable with the curl command:

```
curl -H "Authorization: Bearer $ACCESS_TOKEN"
https://www.googleapis.com/oauth2/v3/userinfo
```
Hands-on with OpenID Connect: Advanced

If you have extra time, visit https://developers.google.com/identity/protocols/OAuth2#libraries for client libraries for programming with OAuth and OpenID Connect.