Partha Bhoumik - 24266012
Mohammad Fahim - 1000055063

Federated Identity Management SAML Implementation using Keycloak

Repository

https://github.com/batcode7/CSE722-Project-2-Federated-Identity-Management-SAML-Implementation-using-Keycloak

System Description

A local federated identity management system built with SAML, where a single Identity Provider (IdP - Keycloak) issues login assertions to two Service Providers (SPs) developed with Flask. The setup demonstrates multi-realm IdP configuration, attribute release, and metadata exchange between the IdP and SPs. At the end, it enables Single Sign-On (SSO), allowing users to log in once and access multiple applications seamlessly.

End-to-End SAML Flow

- Initially the user hits a protected resource on SP1 (/photo in our case). The app sees no session and redirects to /choose-idp.
- Then the user selects IdP-A or IdP-B;
- The browser is then redirected to the IdP (IdP-A for example) SSO endpoint.
- IdP (Idp-A) authenticates the user (Keycloak login) and issues a SAML Response with a signed Assertion containing Subject and AttributeStatement.
- IdP (Idp-A) sends the Response to the SP's ACS via HTTP-POST binding.
- SP1 validates the Response: verifies XML signature with the IdP x509cert, checks audience, recipient, and time conditions.
- On success, SP1 extracts NameID and attributes, stores them in the Flask session , and redirects to the original URL.
- The protected resource now accessed (/photo in our case).
- The user now visits SP2 and tries to access the protected resource. SP2 detects no local session and redirects to /choose-idp.
- The user selects IdP-A and browser redirects to Keycloak IdP-A SSO URL.
- Keycloak checks its realm session cookie and finds an existing, valid SSO session for the user in realm idp-a. IdP-A immediately issues a new SAML Response for SP2 with a signed Assertion.
- SP2 validates the Response, creates a local session, and redirects back to /photo.
- The protected photo is returned, user achieved SSO across SP1 and SP2 via the same IdP session.

Components

- **IdP:** Keycloak 26.3.3
- **Realms:** idp-a and idp-b (for demonstrating multiple IdPs).
- **Users:** who actually have ids in idp-a and idp-b
- **SPs:** Two flask apps (sp1, sp2)

Development Steps & How to run

Follow the readme.md file for setup and How to develop from scratch.md file to build everything from scratch.