Configure Shibboleth SP to Request Multi-Factor Authentication (Duo MFA)

- Introduction
- Before You Begin
- Configure Shibboleth SP
- Important Notes

Introduction

The University of California, San Francisco uses Duo MFA to implement multi-factor authentication for certain applications and users. Multi-factor authentication improves the security of a resource by requiring more than just a username and password in order to gain access to a resource. Check out this article to learn more about Duo MFA.

The MyAccess single-sign-on system is integrated with the Duo MFA system. However, it does not require users to use multi-factor authentication unless an application requests it. Applications can request that the MyAccess single-sign-on system require the additional security of a MFA login by changing parts of the SAML Request the application sends to the MyAccess Identity Provider (IdP). More specifically, the SAML Request must include an "AuthnContextClassRef" element with a value of "https://refeds.org/profile/mfa", which is defined by the REFEDS (Research and Education FEDerationS) group. Read up on the REFEDS MFA assurance profile and the associated specifications and assumptions for more details.

These instructions show how to configure the Shibboleth SP (version 2.5 or later) to include that element in the SAML Request to the MyAccess IdP. These instructions probably will not work with other SAML SP software.

Before You Begin

Before you begin, you must have a working Shibboleth SP configuration that is already integrated with the MyAccess login system. See the installation and configuration instructions for more setup details.

Configure Shibboleth SP

1. Backup the shibboleth2.xml configuration file in the Shibboleth SP working directory.
2. Open the shibboleth2.xml configuration file for editing. Locate the "<ApplicationDefaults..." XML element.
3. Within that XML block, locate the "<Sessions..." XML block, then the "<SSO..." XML block within that and comment out the "<SSO..." block by putting "<!--" before it and "-->" after it.
4. Locate the "<Logout..." block and comment it out in the same way.
5. Add the following "<SessionInitiator..." and metadata definition XML blocks above the commented out "<SSO..." block. Be sure to put the correct MyAccess IdP entityID for the "entityID" parameter value. The staging IdP entityID value is shown in the below example. The production MyAccess IdP entityID (either "https://dp.ucsf.edu/idp/shibboleth" or "urn:mace:incommon:ucsf.edu") can be used as well if your application is already using the production IdP.
shibboleth2.xml SessionInitiator element

```xml
<SessionInitiator type="Chaining"
    Location="/Login"
    isDefault="true"
    id="Login"
    entityID="https://idp-stage.ucsf.edu/idp/shibboleth"/>

<SessionInitiator type="SAML2"
    authnContextClassRef="https://refeds.org/profile/mfa"/>

<SessionInitiator type="Shib1"
    authenticationMethod="https://refeds.org/profile/mfa"/>
</SessionInitiator>

<md:AssertionConsumerService Location="/SAML2/POST" index="1"
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
<md:AssertionConsumerService Location="/SAML2/POST-SimpleSign" index="2"
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"/>
<md:AssertionConsumerService Location="/SAML2/Artifact" index="3"
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>
<md:AssertionConsumerService Location="/SAML2/ECP" index="4"
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS"/>
<md:AssertionConsumerService Location="/SAML/POST" index="5"
    Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post"/>
<md:AssertionConsumerService Location="/SAML/Artifact" index="6"
    Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01"/>

<LogoutInitiator type="Chaining" Location="/Logout">
    <LogoutInitiator type="SAML2"/>
    <LogoutInitiator type="Local"/>
</LogoutInitiator>

<md:SingleLogoutService Location="/SLO/SOAP"
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>
<md:SingleLogoutService Location="/SLO/Redirect"
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"/>
<md:SingleLogoutService Location="/SLO/POST"
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
<md:SingleLogoutService Location="/SLO/Artifact"
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>
<md:ArtifactResolutionService Location="/Artifact/SOAP" index="1"
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>
```

6. The entire shibboleth2.xml configuration file should look something like the example below.

```xml
<SPConfig xmlns="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:conf="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    clockSkew="180">

    <ApplicationDefaults entityID="https://yoursite.ucsf.edu/shibboleth"
        REMOTE_USER="eppn persistent-id targeted-id">

        <Sessions lifetime="28800"
            timeout="3600"
            relayState="ss:mem"
            checkAddress="false"
            handlerSSL="true"
            cookieProps="https"/>
    
```
<SessionInitiator type="Chaining"
Location="/Login"
isDefault="true"
id="Login"
entityID="https://idp-stage.ucsf.edu/idp/shibboleth" />

<SessionInitiator type="SAML2"
authnContextClassRef="https://refeds.org/profile/mfa" />
<SessionInitiator type="Shib1"
authnContextClassRef="https://refeds.org/profile/mfa" />

</SessionInitiator>

<md:AssertionConsumerService Location="/SAML2/POST" index="1"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
<md:AssertionConsumerService Location="/SAML2/POST-SimpleSign" index="2"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"/>
<md:AssertionConsumerService Location="/SAML2/Artifact" index="3"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>
<md:AssertionConsumerService Location="/SAML2/ECP" index="4"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS"/>
<md:AssertionConsumerService Location="/SAML/POST" index="5"
Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post"/>
<md:AssertionConsumerService Location="/SAML/Artifact" index="6"
Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01"/>

<LogoutInitiator type="Chaining" Location="/Logout">
<LogoutInitiator type="SAML2" />
<LogoutInitiator type="Local" />
</LogoutInitiator>

<md:SingleLogoutService Location="/SLO/SOAP"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>
<md:SingleLogoutService Location="/SLO/Redirect"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"/>
<md:SingleLogoutService Location="/SLO/POST"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
<md:SingleLogoutService Location="/SLO/Artifact"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>

<md:ArtifactResolutionService Location="/Artifact/SOAP" index="1"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>

<!--
<SSO entityID="https://idp-stage.ucsf.edu/idp/shibboleth" >
  SAML2 SAML1
</SSO>
-->

<!--
<Logout>SAML2 Local</Logout>
-->

<Handler type="MetadataGenerator" Location="/Metadata" signing="false"/>
<Handler type="Status" Location="/Status" acl="127.0.0.1 ::1"/>
<Handler type="Session" Location="/Session" showAttributeValues="false"/>
<Handler type="DiscoveryFeed" Location="/DiscoveryFeed"/>

</Sessions>

<Errors supportContact="YOUR.NAME@ucsf.edu"
helpLocation="/about.html"
styleSheet="/shibboleth-sp/main.css"/>

<MetadataProvider type="XML" validate="true" file="idp-metadata.xml"/>
<AttributeExtractor type="XML" validate="true" reloadChanges="false" path="attribute-map.xml"/>
<AttributeResolver type="Query" subjectMatch="true"/>
<AttributeFilter type="XML" validate="true" path="attribute-policy.xml"/>
<CredentialResolver type="File" key="sp-key.pem" certificate="sp-cert.pem"/>

</ApplicationDefaults>
7. Restart the Shibboleth SP daemon (a.k.a service in Windows) as well as the web server (Apache or IIS).
8. Now all web browser requests to access your Shibboleth SP protected web site will cause the MyAccess IdP to initiate multi-factor authentication by first asking the user to login with their username and password (generally, their AD username and password), then prompt them to use Duo MFA to verify their identity. If they are not currently enrolled in Duo MFA, they will not be able to access your application.

**Important Notes**

- **If an application like the one above requests multi-factor authentication, then ALL users who use MyAccess to login to that application must use Duo MFA** to gain access to that application. It's not possible to exempt some users from using MFA while requiring MFA for other users. However, the Shibboleth SP software does allow you to configure separate applications within the same host (such as subdirectories in an existing web site) which can be configured to require MFA separately from the default application. This advanced topic is not covered in this document. You can read more about the "ApplicationOverride" configuration option with the Shibboleth SP software on the Shibboleth wiki site.

- **Duo MFA uses the username that the user types into the MyAccess login page to process multi-factor authentication.** If users use a username to login to MyAccess that is not already enrolled into Duo MFA, they'll be denied access and will not be able to access your application.

**EXAMPLE:** Let's say you're enrolled in the Duo MFA system with the "jdoe" Active Directory username, but you usually login to MyAccess with your old SF ID of "SF123456". If you login to MyAccess using your SF123456 username and associated password, then attempt to access the above application protected by the Shibboleth SP software that requests multi-factor authentication, you'll be denied access during the Duo authentication process, even though you are already enrolled into Duo with your "jdoe" AD account. If you want to use Duo MFA with your AD account, you must login to MyAccess with the same "jdoe" username and password before going to a MFA protected SSO application.