Configure a Service Provider for Two-Factor Authentication

Purpose

This guide shows how to configure a Shibboleth Service Provider (SP) to initiate two-factor authentication for all its Shibboleth logins. If you need to mix password-based logins and two-factor logins for different content, follow Configure a Service Provider for Step-up Two-Factor Authentication instead.

Background

The UW Shibboleth Identity Provider (IdP) supports two-factor authentication using Duo 2-Factor Authentication (2FA)

If your SP accepts authentication assertions from other IdPs, you should be aware that two-factor authentication may not be supported by all IdPs.

Configuration

Two-factor authentication can be configured in several ways, each with different implications. All methods depend on specifying an "authnContextClassRef" attribute with a particular value.

- https://refeds.org/profile/mfa - This value is the current standard and is used in all examples below.
- urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken - This value was used for years but it is now deprecated.

The available configuration options differ in where the "authnContextClassRef" attribute is specified in the configuration file.

Configure the service provider to request 2FA

1. The first example requests two-factor login by modifying the default <SSO> element. This configuration will force all Shibboleth logins from this SP to use two-factor. If Shibboleth has been configured for multiple web sites (virtual hosts), two-factor will apply to all web sites.

   <SSO entityID="urn:mace:incommon:washington.edu"
        authnContextClassRef="https://refeds.org/profile/mfa">
        SAML2
   </SSO>

2. The second example requests two-factor login by modifying a <Host> element. This configuration will force all Shibboleth logins from the specified web site (virtual host) to use two-factor. If Shibboleth has been configured for multiple web sites (virtual hosts), two-factor will only apply only to the specified web site. Note that the requirement for a Shibboleth session and the requirement for two-factor can be specified in the <Host> statement. In this example, all requests for content from site2 will require a Shibboleth session with two-factor.

   <Host name="site1.washington.edu" authType="shibboleth" requireSession="true" />
   <Host name="site2.washington.edu"
        authnContextClassRef="https://refeds.org/profile/mfa"
        authType="shibboleth" requireSession="true" />
3. The third example requests two-factor login by modifying a `<Path>` element. This configuration will force Shibboleth logins for content in the specified path to use two-factor. Note that the requirement for a Shibboleth session and the requirement for two-factor can be specified in the `<Path>` statement. In this example, requests for content in the `https://site1.washington.edu/twofactor/` directory will require a Shibboleth session with two-factor.

```xml
<Host name="site1.washington.edu">
  <Path name="twofactor" authType="shibboleth" requireSession="true"
    authnContextClassRef="https://refeds.org/profile/mfa">
  </Path>
</Host>
```

If this site also has content that only requires a Shibboleth session with password authentication, this configuration will not work reliably. If the user requests content in the `/twofactor` directory first, two-factor will be enforced. If the user requests content that only requires password authentication and then requests content in the `/twofactor` directory, two-factor will not be enforced. The existing Shibboleth session based on password authentication will not be upgraded to two-factor. If you need to mix password-based logins and two-factor logins for different content, follow Configure a Service Provider for Step-up Two-Factor Authentication instead.

4. The fourth example requests two-factor login at the UW IdP for the session with id="uwtoken" by modifying the appropriate `<SessionInitiator>` element:

```xml
<!-- Token auth at UW -->
<SessionInitiator type="Chaining" Location="/SecureLogin" isDefault="false" id="uwtoken"
  relayState="cookie" entityID="urn:mace:incommon:washington.edu"
  authnContextClassRef="https://refeds.org/profile/mfa">
  <SessionInitiator type="SAML2" template="bindingTemplate.html"/>
</SessionInitiator>
```

Configure the service provider to verify 2FA was used in response

The process to configure the service provider to verify 2FA was used during sign-in depends on the platform used.

**Apache 2.4**

To configure Apache 2.4 to require the authentication was completed with 2FA a combination of Shibboleth specific and standard Apache access control directive can be used. The below example will restrict access to any users who have used 2FA to sign in with their UW NetID.

These Apache directives should be located either within the Apache configuration itself or in `.htaccess` files within in specific directories. The choice between the two depends on how Apache is configured.

```xml
AuthType shibboleth
ShibRequestSetting requireSession true

<RequireAll>
  Require authnContextClassRef https://refeds.org/profile/mfa
  Require shib-session
</RequireAll>
```

**Apache 2.2**

Unfortunately Apache 2.2 does not support the `<RequireAll>` block and interprets multiple Require directives with an implicit 'OR'. Shibboleth SP instead provides an equivalent functionality to RequireAll. **Note:** The ShibRequireAll directive is NOT compatible with Apache 2.4.

```xml
AuthType shibboleth
ShibRequestSetting requireSession true
ShibRequireAll on
ShibCompatWith24 on

Require authnContextClassRef https://refeds.org/profile/mfa
Require shib-session
```

**Windows IIS**

IIS does not support this checking within the web server's configuration. Instead, access control will need to be configured within the `<RequestMapper>` in `shibboleth2.xml`. 

The placement of the `<AccessControl>` elements and their children will depend on which configuration above is utilized. The below example references example 3. It can also be placed within the `<Host>` element if example 2 is used.

See the documentation from the Shibboleth Project for more information: https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAccessControl and https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPRequestMap

```xml
<RequestMapper type="Native">
  <RequestMap>
    <Host name="your-host">
      <Path name="twofactor" authType="shibboleth" requireSession="true" authnContextClassRef="https://refeds.org/profile/mfa">
        <AccessControl>
          <Rule require="authnContextClassRef">https://refeds.org/profile/mfa</Rule>
        </AccessControl>
      </Path>
    </Host>
  </RequestMap>
</RequestMapper>
```

Additionally, after authentication your application’s session environment will include specific attributes indicating the authentication method actually used by the IdP. These can be used as an additional verification mechanism, instead or in addition to using the web server methods above.

On Apache:

- Shib_Authentication_Method: https://refeds.org/profile/mfa
- Shib_AuthnContext_Class: https://refeds.org/profile/mfa

On IIS:

- HTTP_SHIBAUTHENTICATIONMETHOD: https://refeds.org/profile/mfa
- HTTP_SHIBAUTHNCONTEXTCLASS: https://refeds.org/profile/mfa

**Recommendations**

If your intent is to enforce two-factor authentication for all Shibboleth sessions, you should probably use the `<SSO>` element. If you have multiple websites on a host configured for Shibboleth and you only want the two-factor requirement enforced on some of the web sites you should probably use the `<Host>` element.

**Single Sign-On**

If you wish to always enforce a new two-factor login for every visitor to your site, you should add `forceAuth="true"` to the block of XML where you configured "authnContextClassRef". Otherwise, if a user has already signed in with 2FA during their SSO session they will not need to re-authenticate. See Configure a Service Provider to Force Re-Authentication for more information.

**See Also**

- Configure a Service Provider for Step-up Two-Factor Authentication
- Shibboleth Project's SessionInitiator reference