Duo 2FA On Demand

Configuring Duo authentication for use upon request

Triggering Duo, or any authentication method, by request involves naming that method, including that method in the IdP configuration and then requesting it from the SP.

Authentication Context

SAML V2.0 introduced the notion of an *Authentication Context Class*, denoted by an URI. A particular Authentication Context Class may address a wide range of assurance issues including user registration, identity proofing and credentialing, in addition to the method of authentication.

Naming the method

Since username/password authentication is the norm, an IdP usually asserts the following Authentication Context Class URI:

urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport

The above Authentication Context Class is defined by the OASIS SAML specification, which is why the URI has stem "urn:oasis". To define your own Authentication Context Class, begin by choosing an appropriate URI. For example, the URI selected for use at USC is:

urn:usc.edu:ac:classes:PasswordProtectedTransport:duo

Alternatively, choose a URL in your institution's namespace, that is, a URL rooted in a domain owned by your institution. (In this case, the URL is just a name, so it need not resolve to an actual web resource.)

Now configure the chosen Authentication Context Class URI into the Duo login handler in the handler.xml of the IdP. After following the Duo instructions to achieve a basic install, you should have a LoginHandler of type two factor:TwoFactorLogin. Normally the login handler would be configured like this:

```
<ph:AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</ph:
AuthenticationMethod>
```

Change this to:

```
<ph:AuthenticationMethod>urn:usc.edu:ac:classes:PasswordProtectedTransport:duo</ph:AuthenticationMethod>
```

Use the Authentication Context Class URI chosen for this purpose.

Modify servlet parameters:

The default configuration in web.xml looks like this:

```
   <servlet>
     <servlet-name>TwoFactorLoginHandler</servlet-name>
     <servlet-class>com.duosecurity.shibboleth.idp.twofactor.TwoFactorLoginServlet</servlet-class>
     <load-on-startup>4</load-on-startup>
   </servlet>
```

To configure this servlet to handle only the specific method intended, add the `authnMethod` init param like so:

```
   <servlet>
     <servlet-name>TwoFactorLoginHandler</servlet-name>
     <servlet-class>com.duosecurity.shibboleth.idp.twofactor.TwoFactorLoginServlet</servlet-class>
       <init-param>
           <param-name>authnMethod</param-name>
           <param-value>urn:usc.edu:ac:classes:PasswordProtectedTransport:duo</param-value>
       </init-param>
     <load-on-startup>4</load-on-startup>
   </servlet>
```

SP Request for particular authentication method

There are several ways to cause the SP to cause its initial request to contain the parameter of the desired authentication method. The easiest way to make this the default is to configure this on the SessionInitiator element for the app in the shibboleth2.xml file.

The SessionInitiator is extremely configurable and can allow for many customizations and different ways of using it. In the latest releases, this element is not configured by default and the software automatically configures it to the defaults. The following line can be added to the Sessions element, or it can be used inside of another SessionInitiator of type="Chaining" if you have multiple ways to initiate a session. In any case, the key attribute is authnContextClassRef. This attribute is configured on the SessionInitiator of type SAML2. For example:

```
<SessionInitiator type="SAML2" Location="/Login" isDefault="true" template="bindingTemplate.html"
authnContextClassRef="urn:usc.edu:ac:classes:PasswordProtectedTransport:duo" forceAuthn="true"/>
```

It is good practice to use forceAuthn as well in this scenario of increased security so that the IdP will not allow the reuse of an existing authentication.

Ref: https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPSessionInitiator

Query string session initation

Many of the configurable options on the SessionInitiator can be invoked by a login from a specially crafted query string. This method can be used as a redirect from within the app when login is needed. For example:

```
https://sp.example.org/Shibboleth.sso/Login?forceAuthn=true&authnContextClassRef=urn:usc.edu:ac:classes:
PasswordProtectedTransport:duo&target=https%3A%2F%2Fsp.example.org%2Fresource.asp
```

Ref: https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPSessionCreationParameters

IdP default authentication method

It is more reliable to make sure the SP requests the method each time, whenever appropriate, but you can also specify a default at the IdP. This is not enforced, but only used as a suggestion when no specific method was requested. To configure this default suggested authentication method, edit the relying-party.xml definition for the application. You may need to make a custom entry separate from the default definition or definitions that cover a group of applications in a metadata file. Once this definition is in place, use the following attribute on the RelyingParty element:

defaultAuthenticationMethod="urn:usc.edu:ac:classes:PasswordProtectedTransport:duo"

Ref: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPRelyingParty