IntegrateWithLDAP

LDAP Groups Integration

The restriction of access to a resource to a specific set of users can be implemented in a variety of ways. Traditionally, this has been done with standard LDAP groups, and there may be a need or desire to continue using these even as attribute-based groups are supported. There are several ways this integration can be performed. In a situation where there is no need to integrate with LDAP, there is a small number of users, or the users are scattered amongst a large number of IdPs, it's recommended that an AuthGroupFile be used.

There are two ways integration can be achieved:

1. Import the group information using the IdP, transport it using an appropriate attribute name, and export it as `memberof` using the following as an example:

AAP.xml:

```
<AttributeRule Name="urn:mace:example.org:attributes:group" Header="MEMBEROF" Alias="memberof">
        <SiteRule Name="urn:mace:example.org:SSO">
                <AnyValue/>
        </SiteRule>
</AttributeRule>
```

Release of this information in an interrealm deployment is dangerous and makes little sense. eduPersonEntitlement or custom attributes should be used instead.

ARP.xml:

```
<Rule>
        <Target>
                <Requester>urn:mace:example.org:SSO</Requester>
        </Target>
        <Attribute name="urn:mace:example.org:attributes:group">
                <AnyValue release="permit"/>
        </Attribute>
</Rule>
```

resolver.xml

```
<SimpleAttributeDefinition id="urn:mace:example.org:attributes:group">
        <DataConnectorDependency requires="directory"/>
</SimpleAttributeDefinition>
```

.htaccess

```
<Location /topsecret>
  AuthType shibboleth
  ShibRequireSession On
  Require memberof "My Service Users"
</Location>
```

2. Alternatively, assuming a few limitations on how the LDAP module works (specifically, it's at least implemented with `r->user`), it's possible to actually use the LDAP module itself (or any other auth/z module) for the auth/z and access control once Shibboleth transports the information. Everything from Shibboleth in the above example remains, but not the Apache `AuthType` or `Require` statements. These change based on the implementation of the auth /z module in question.