WebAC Testing: Creating Resources and Authorizations

These curl commands will create a small set of test resources that can be used to test various authorization scenarios. The resources and ACLs can be also modified to test other features of the WebAC implementation not covered by this set. As they stand, this sample set of resources tests the following features of the F4 WebAC implementation phase1:

- This test repository uses PUT to create a tree of named resources for the sake of simplicity and clarity. In a typical real-world implementation, F4 would most likely be allowed to create resources at arbitrary locations in the JCR tree, and those resources would be linked through RDF properties (according to PCDM or LDP containment) rather than the manually constructed hierarchy used here. This manual construction method takes advantage of a feature of F4, which automatically connects resources to their children with an ldp:contains property.
- The top level of this sample set contains a single collection (collection1) and three ACLs. Collection1 in turn contains Item1 and Subcollection1, and the latter contains two additional items (item2 and item3).
- Collection1 is protected by ACL1, which grants Read/Write access to "adminuser" and Read access to "testuser". These access permissions should apply down the tree to any resource not covered by its own ACL (including subcollection1 and item2).
- Item1, which is also a child of Collection1, is protected by its own ACL (ACL2), which is designed to test the principle that the more permissive authorization wins in cases where more than one authorization might apply to a given request. ACL2 consists of two authorizations, one that grants Read/Write permission to adminuser, and another which grants only Read permission to adminuser. This scenario would not likely come about in a real world implementation except possibly through error, but it serves here as a stand-in for a conflict between, for example, two applicable agent classes (a single user who is a member of both class A and class B, and those classes have different authorizations for a given resource).
- Finally, item3 has its own ACL (ACL3), which grants only Read permission to adminuser. This resource can be used to test how the first ACL found is the applicable ACL (i.e. F4 will not continue looking for ACLs up the tree, and thus ACL1, which applies to Collection1 and by extension its children, is preempted by the more restrictive ACL3, which is found first in the algorithm for finding the applicable ACL (see "Steps in Determining the Effective Authorization" in the WebAC documentation). A more permissive ACL could just as easily be created to allow more access to some subset of a restricted collection.
- Having created the resources below, navigate to them in a browser (using different incognito/private tabs/windows to authenticate as testuser: password1 and adminuser:password2). The Fedora REST interface will tell you whether it recognizes the user account in question as having read (displaying the properties on the left, but not the update column on the right), read/write (displaying both columns), or no permission (403 forbidden) to the page you have requested. Alternatively, you can test via curl using GET requests to read, and PATCH requests to add properties to pages where the user in question should have write permissions.

Create collection1, subcollection1, item1, item2, item3

```
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/collection1"
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/collection1/subcollection1"
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/collection1/item1"
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/collection1/subcollection1/item2"
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/collection1/subcollection1/item3"
```

Create ACL1, add authorizations 1 & 2

```
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/acl1"
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/acl1/auth1"
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/acl1/auth2"
curl -i -u fedoraAdmin:secret3 -X PATCH -H "Content-Type: application/sparql-update" -d "PREFIX acl:
<http://www.w3.org/ns/auth/acl#> INSERT { <> a acl:Authorization; acl:accessTo </fcrepo/rest/collection1>; acl:
mode acl:Read, acl:Write; acl:agent 'adminuser' . } WHERE {}" http://localhost:8080/fcrepo/rest/acl1/auth1
curl -i -u fedoraAdmin:secret3 -X PATCH -H "Content-Type: application/sparql-update" -d "PREFIX acl:
<http://www.w3.org/ns/auth/acl#> INSERT { <> a acl:Authorization; acl:accessTo </fcrepo/rest/collection1>; acl:
mode acl:Read; acl:agent 'testuser' . } WHERE {}" http://localhost:8080/fcrepo/rest/acl1/auth2
```

Create ACL2, add authorizations 3 & 4

```
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/acl2"
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/acl2/auth3"
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/acl2/auth4"
curl -i -u fedoraAdmin:secret3 -X PATCH -H "Content-Type: application/sparql-update" -d "PREFIX acl:
<http://www.w3.org/ns/auth/acl#> INSERT { <> a acl:Authorization; acl:accessTo </fcrepo/rest/collection1
/item1>; acl:mode acl:Read, acl:Write; acl:agent 'adminuser' . } WHERE {}" http://localhost:8080/fcrepo/rest
/acl2/auth3
curl -i -u fedoraAdmin:secret3 -X PATCH -H "Content-Type: application/sparql-update" -d "PREFIX acl:
<http://www.w3.org/ns/auth/acl#> INSERT { <> a acl:Authorization; acl:accessTo </fcrepo/rest/collection1
/item1>; acl:mode acl:Read; acl:agent 'adminuser' . } WHERE {}" http://localhost:8080/fcrepo/rest/acl2/auth4
```

Create ACL3, add authorization 5

```
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/acl3"
curl -i -u fedoraAdmin:secret3 -X PUT "http://localhost:8080/fcrepo/rest/acl3/auth5"
curl -i -u fedoraAdmin:secret3 -X PATCH -H "Content-Type: application/sparql-update" -d "PREFIX acl:
<http://www.w3.org/ns/auth/acl#> INSERT { <> a acl:Authorization; acl:accessTo </fcrepo/rest/collection1
/subcollection1/item3>; acl:mode acl:Read; acl:agent 'adminuser' . } WHERE {}" http://localhost:8080/fcrepo/rest
/acl3/auth5
```

Point collection1 to ACL1, item1 to ACL2, and item3 to ACL3

```
curl -i -u fedoraAdmin:secret3 -X PATCH -H "Content-Type: application/sparql-update" -d "PREFIX acl:
<http://www.w3.org/ns/auth/acl#> INSERT { <> acl:accessControl <http://localhost:8080/fcrepo/rest/acl1> . }
WHERE {}" "http://localhost:8080/fcrepo/rest/collection1"
curl -i -u fedoraAdmin:secret3 -X PATCH -H "Content-Type: application/sparql-update" -d "PREFIX acl:
<http://www.w3.org/ns/auth/acl#> INSERT { <> acl:accessControl <http://localhost:8080/fcrepo/rest/acl2> . }
WHERE {}" "http://localhost:8080/fcrepo/rest/collection1/item1"
curl -i -u fedoraAdmin:secret3 -X PATCH -H "Content-Type: application/sparql-update" -d "PREFIX acl:
<http://www.w3.org/ns/auth/acl#> INSERT { <> acl:accessControl <http://localhost:8080/fcrepo/rest/acl3> . }
WHERE {}" "http://localhost:8080/fcrepo/rest/collection1/subcollection1/item3"
```