UC3Merritt

University of California Curation Center

Merritt Repository LDAP-Based Access Control

Rev. 0.9 – 2012-06-04

1 Introduction

Currently, access to digital content managed in the Merritt curation repository is limited to registered and logged-in users only. These users are granted read-only or read/write access to content on a per-collection basis. While a comprehensive and granular role-based access control mechanism is the long-term development goal [3], a number of high-priority, incremental enhancements should be pursued in the interim. These include (in priority order):

- Anonymous read-only access to designated collections.
- Stable (and bookmarkable and resolvable) URLs for collection/object/version landing pages and files.
- Click-through data user agreements (DUAs) as a precondition to access to designated collections.
- Meaningful distinction in access control granularity between object data and metadata.
- Limited time embargoes on access to designated collections.

NOTE The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are to be interpreted as described in RFC 2119 [2].

2 LDAP

The current Merritt access control mechanism is based on the following collection-centric LDAP information model [5]:

```
ou=uc3,dc=cdlib,dc=org
   ou=mrt-classes
      ou=<collection>
         cn=read
            uid=<user>; …
         cn=write
            uid=<user>; …
   ou=People
      uid=<user>
```

Individual users are granted appropriate access privileges by being associated with the “read” and/or write” common name (cn) for a particular collection.

3 Anonymous access

The simplest incremental change necessary to provide collection level anonymous access is to define a new “anonymous” user:

```
ou=uc3,dc=cdlib,dc=org
   ou=People
      uid=anonymous
         arkID=ark:/99166/p9<anonymous-user-ark>
         cn=anonymous
         objectClass=inetOrgPerson; Auxillary: merrittUser
         sn=anonymous
```

This “anonymous” user can then be added to the “read” common name (cn) of collections for which anonymous access is appropriate. The selection of this user account can be handled in two ways, one simple, but with a non-optimal user experience; the other more complex, but with a smoother user experience.

If the “anonymous” user has explicit “read” or “write” access permission to a collection, then *all* users have implicit “read” or “write” access as well. For example, for the following collection and users, named users “abc” and “def” both have read access to the collection: “abc” by virtue of “abc” being associated with the “read” CN and “def” by virtue of “anonymous” being associated with the “read” CN.

```
ou=<collection>
   cn=read
      uid=anonymous; abc
ou=People
   uid=anonymous
      ...
   uid=abc
      ...
   uid=def
      ...
```

Whenever Merritt needs to determine read and read permissions, it should do so in a two step manner:

- Determine whether or not the “anonymous” user has explicit permission. If yes, then allow the action; if not, then:
- Determine whether or not the named user has explicit permission. If yes, then allow the action; if not, deny the action.

3.1 Explicit anonymous user

The simplest way to specify the “anonymous” user is through a small modification of the Merritt Login page [http://merritt-stage.cdlib.org/login], offering a choice between a “guest” login with restricted read-only privileges and the existing credentialed login (see Figure 1).

The response to clicking the “Guest” button is to define the current user as the “anonymous” user and start an appropriate UI session. The behavior of the “Login” button SHALL remain unchanged. No other modifications to the Merritt UI would be necessary; the “anonymous” user will automatically have all privileges granted to any other read-only user role.

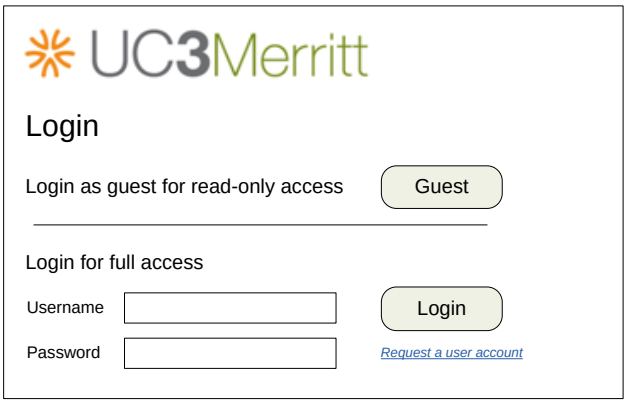

Figure 1 – Login challenge screen

3.2 Implicit anonymous user

While the "guest" login option is effective, it does require an affirmative action by the user. A smoother user experience could be provided by making the assumption of the "anonymous" user the default behavior in the absence of an explicit login using username/password credentials. Note, however, that this will require a more substantial reworking of the Merritt UI, particularly with respect to the following UI pages:

- Welcome page.
- Collection landing page.
- Object landing page.
- Version landing page.
- "Change collection" menu item.
- "Add object" menu item.

[*Additional analysis necessary*]

4 Stable URLs

All of the UI pages for major information resources and actions in the Merritt MUST have stable URLs that are bookmarkable and resolvable. Under the assumption that these URLs will be circulated widely, for consumption by both machines and people, they SHOULD be short and easily transcribed; any abbreviated syntactical constructs internal to the URL SHOULD have reasonable mnemonic value. The general pattern of the URLs is:

http://merritt.cdlib.org/*mode*/*collectionid*|*objectid*[/*versionid*[/*fileid*]]

where *mode* declares the underlying action: "a" for add (or submit), "d" for download, "m" for metadata, or "s" for search (or lookup); *collectionid* is a collection identifier; *objectid* is an object

identifier, i.e., its ARK; *versionid* is a version number; *fileid* is a file pathname; a vertical bar | indicates alternative choices; and brackets [and] indicate optional elements. The following table defines the current and newly proposed forms of all UI URLs. For notational convenience the URLs marked with * are structured as if for GET requests. In actuality, these links are serviced through POST requests and the query string parameters are passed in the request body.

Page	***URL* [*current* / *proposed*]**
Home page	`./home`
Login page	`./login`
Welcome page	`./home/choose_collection`
Collection landing page	`./collection?group=`*`collectionid`*
	`./m/`*`collectionid`*
Object landing page	`./object?group=`*`collectionid`*`&object=`*`objectid`*
	`./m/`*`objectid`*
Object download *	`./object/download?group=`*`collectionid`*`&object=`*`objectid`*
	`./d/`*`objectid`*
Version landing page	`./version?group=`*`collectionid`*`&object=`*`objectid`*`&version=`*`versionid`*
	`./m/`*`objectid/versionid`*
Version download *	`./version/download?group=`*`collectionid`*`&object=`*`objectid`*`&version=`*`versionid`*
	`./d/`*`objectid/versioned`*
File download	`./file/display?group=`*`collectionid`*`&object=`*`objectid`*`&version=`*`versionid`*`&file=`*`fileid`*
	`./d/`*`objectid/versionid/fileid`*
Lookup	`./collection/search_results?group=`*`collectionid`*`&terms=`*`terms`*
	`./s/`*`collectionid`*`?term=`*`terms`*
Add object page	`./object/add?group=`*`collectionid`*
	`./a/`*`collectionid`*
Logout page	`./logout`
Help	`./help/…`
Documentation	`./doc/…`
Image credits	`./home/credits`

Table 1 – UI page URLs

All URL reserved characters occurring in the *objectid* or *fileid* components of UI URLs MUST be URL encoded [1]. For example:

Incorrect: http://merritt.cdlib.org/d/ark:/99999/fk4ab/n34kq/1/ocr/p017.txt
Correct: http://merritt.cdlib.org/d/ark%3a%2f99999%2ffk4ab%3fn34kq/1/ocr%3fp017.txt

Note that care must be taken to distinguish between URLs for a collection and object landing page, which share the same general syntax: http://www.cdlib.org/m/*id*, where *id* can be either a collection or object identifier.

As previously noted, the URLs for object and version download in Table 1 are formatted as if those operations were initiated through an HTTP GET request; that is, they are in the conventional URL form with a path and query string. However, these two operations are actually initiated through a POST request. The URL to which the POST is directed corresponds to the path part of the URL; the query string parameters are passed in the body of the POST request. Fortunately, the Rails framework [3] underlying the Merritt UI abstracts away the distinction between GET and POST. Actually supplying the Table 1 form in the browser address bar (thus simulating a GET request) initiates the same operation as a POST request.

4.1 Inbound links

Resolving an inbound link to one of these pages before an authenticated session has been established MUST result in an automatic redirect to the login page, and following a successful login, either as a guest or a credentialed user, MUST be followed by an automatic redirect back to the original page (or an error page if the user does have the appropriate privilege to access the original page). However, if that page is associated with a collection designated for anonymous read-only access (i.e., if the "anonymous" user is associated with the "read" common name), the redirect to the login page MUST NOT occur; rather, the user SHALL have direct access to the page without any login challenge, and without establishing an authenticated session.

4.2 Permalinks

The object landing page MUST display a preferred "permalink" based on the object primary ARK identifier in N2T resolvable form: "http://n2t.net/ark: /*naan*/*identifier*". This could, for example, be presented at the top of the object metadata (Figure 2):

Cruse, Patricia. **UC3 presentation for Computer History Museum**

permanent link:	http://n2t.net/ark:/99999/fk4tq6h62
primary identifier:	ark:/99999/fk4tq6h62
title:	UC3 presentation for Computer History Museum
creator:	Cruse, Patricia
…	…

Figure 2 – Permalinks

5 Metadata/data-grained control

The simplest incremental change necessary to support a meaningful access control distinction between object data and metadata is to redefine the semantics of the current collection-level actions in the LDAP information model, so that the "read" common name (cn) means "read (metadata)" and a new "download" common name means "read" or "download (data)" . (The semantics of the existing "write" action remain unchanged.)

```
ou=uc3,dc=cdlib,dc=org
   ou=mrt-classes
      ou=<collection>
         cn=read
            uid=<user>; …
         cn=write
            uid=<user>; …
         cn=download
            uid=<user>; …
```

The visibility of collections on the welcome page and the "Change collection" menu item MUST be controlled by the "read" common names in LDAP collection records. In other words, access to collection landing pages is considered to be reading metadata. Links to all objects in a visible collection MUST be supplied on the collection's landing page. The presence of the "Download object" on an object landing page and the "Download version" on a version landing page MUST be controlled by the "download" common name in that object's collection's LDAP record. The actionability of file links on an object or version landing page MUST be controlled by the "download" common name in that object's collection's LDAP record. Users with "read" but not "download" privilege WILL still see the enumeration of the files and their summary metadata (MIME type and size), but they will *not* be actionable links, that is, they will not be formatted as HTML "*file*" tags. In other words, the *fact* that a file is a component of an object is considered (readable) metadata. Note that these same access control decisions MUST be enforced when a user is resolving a bookmarked URL rather than navigating in the UI.

As with "read" and "write" access permissions, if the "anonymous" user has explicit "download" permission, then *all* users have implicit "download" permission as well.

7 Embargoes

[*To be determined*]

References

[1] T. Berners-Lee, R. Fielding, and L. Masinter, *Uniform Resource Identifier (URI): Generic syntax*, STD 66, RFC 3986, January 2005 <http://www.ietf.org/rfc/rfc3986.txt>.

[2] S. Bradner, *Key Words for Use in RFCs to Indicate Requirement Levels*, BCP 14, RFC 2119, March 1997 <http://www.ietf.org/rfc/rfc2119.txt>.

[3] David Heinemeier Hansson, *Ruby on Rails*, 2012 <http://rubyonrails.org/>.

[4] UC Curation Center, *GhOST: Merritt Access Control*, 2011.

[5] K. Zeilenga, *Lightweight Directory Access Protocol (LDAP): Directory Information Models*, RFC 4512, June 2006 <http://www.irtf.org/rfc/rfc4512.txt>.