Infrastructure - Bug #8215

Consider how Subjects are compared (e.g. HTTP vs. HTTPS ORCID URIs)

2017-11-07 01:41 - Bryce Mecum

Status:	New	**Start date:**	2017-11-07
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Milestone:	None	**Story Points:**	
Product Version:			

Description

I know this has come up a few times in the last few years and it has bitten us a lot in day-to-day operations at the Arctic Data Center. Some Subjects in DataONE authentication appear to be compared literally so these two subjects are not considered equivalent:

http://orcid.org/0000-0002-0381-3766
https://orcid.org/0000-0002-0381-3766

even though a reasonable spectator might consider them equivalent.

Where this causes trouble seems to be when System Metadata is authored by users, specifically the rightsHolder and accessPolicy portions, and what's in the System Metadata does not literally match the user's actual Subject. As an example, when a user logs in via ORCID, their DataONE Subject ends up being the *http* variant of their ORCID URI, so if they use the *https* variant of their ORCID URI in their System Metadata, API calls requiring read, write, and changePermission permission fail for them because the literal string comparison determines the two Subjects to non-equivalent.

I think this may only affect ORCIDs right now because Subjects such as LDAP DNs may already be compared in a string-insensitive fashion. Though I'm not sure on this point.

A few of us on the NCEAS dev team discussed this and we think it's fair to compare Subjects more intelligently, or at least be more aware of the semantic structure of the Subject string. This would have numerous benefits, such as:

- Prevent users from becoming hopelessly confused when they can't figure out why they can't read/write Objects (people often don't notice http vs https) which will make DataONE seem friendlier
- Make DataONE authentication less dependent upon protocols such as HTTP/HTTPS, both of which my change (i.e., ORCID may disable HTTPS; HTTP(S) may be replaced by a future web protocol) and thus more future-proof

This type of change would require changes in one or more software projects:

- DataONE Portal or libclient Java (I'm not entirely sure where this check is done right now)
- Architecture documentation
- (Potentially any/all) MN software stacks (At least a review would be needed)
- (Potentially) Client tools (e.g., R, Python) (At least a review would be needed)

History

#1 - 2017-11-07 02:38 - Dave Vieglais

- Tracker changed from Decision to Bug

This is really a client problem. Users should be strongly encourage to use https URLs for identity. This can be achieved by education and by tool implementors checking and warning users when attempting to use http identities.

Implementing a (http) protocol agnostic comparison at the CN level will add complexity and processing for handling an issue which really should be resolved before content enters the system.

Should it become necessary to deploy such a change system wide, implementors should be cautious to not ignore the http protocol (e.g. split on // and use only the second half), since otherwise it would be relatively easy to spoof an id.

#2 - 2017-11-07 18:47 - Bryce Mecum

> Users should be strongly encourage to use https URLs for identity.

Two immediate reactions:

- If we're doing literal string comparisons, these Subjects are not URLs but just strings. Comparison of URLs is subject to a normalization routine. I'm aware this might come off as pedantic.
- I was under the impression Subjects were URIs, so seeing you call them URLs is interesting. While the semantics community is split on what to do about protocols and URIs, a common practice is to consider the protocol to be a separate concern to what is being identified (in this case, and ORCID is being identified).

Second, I hear you suggesting we switch the portal to use HTTPS. Currently, the portal returns an HTTP URI for the identified ORCID. This change, done alone, would:

- Require updating the System Metadata for potentially millions of Objects across DataONE in order to let users continue to see/edit/whatever their content
- Break many client use cases that have hard-coded HTTP URIs for their Subjects

I'm all for trying to doing things right. The safest thing is to keep the status quo, and I think the only alternative is to (1) switch the portal to use HTTPS URIs for ORCIDs and (2) make identity verification across the system tolerant to protocol differences.

#3 - 2017-11-07 19:40 - Dave Vieglais

In this case the subjects are actually urls and uris - though, yes I agree that within the context of DataONE systems, they are to be treated as URIs.

I was not aware the portal is presenting HTTP URIs for ORCIDS. That is absolutely an error and should be corrected, however it seems that as a consequence we now have to live with this mess.

So - yes, it seems the only reasonable path forward is to be agnostic with respect to http vs https as the protocol specifier in the URI.

#4 - 2017-11-07 22:11 - Dave Vieglais

Including for implementor guidance.

In a nutshell, "http" and "https" are different schemes, and so technically a comparison of otherwise identical URIs should indicate the URIs are not the same. From a practical perspective though the ORCID URIs are otherwise equivalent. Because ignoring the difference between http and https schemes does raise security concerns, the final implementation should not blithely ignore the "host" portion of the URI, but should take into consideration known exceptions to rule that "http" and "https" schemes are not equivalent. For example, in this case, it is known that for the host "orcid.org" the http and https schemes may be treated as identical. This should not be considered a generalization for all http and https URI schemes however.

RFC 3986[1] along with RFC 6874, RFC 7320 provides the specifications for URI syntax.

The pertinent part of the specification is provided in § 3.1 wherein "scheme" is defined as:

```
scheme      = ALPHA *( ALPHA / DIGIT / "+" / "-" / "." )
```

with additional notes that "...schemes are case-insensitive, the canonical form is lowercase and documents that specify schemes must do so with lowercase letters."

Hence, under RFC 3986, "http" is equivalent to "HTTP", however "https" is not equivalent to "http".

RFC 3986 § 6 provides the rules for normalization and comparison of URIs. §§ 6.2.3 describing scheme-based normalization and 6.2.4 protocol-based normalization.

RFC 3896 § 7 provides discussion on security considerations which should also be considered by implementors.

[1] https://tools.ietf.org/html/rfc3986