Subject API security by realm

Different users might have different privacy requirements for the Subject API. Security by realm is a new feature in 2.2 which is implemented in the JDBC2 source adapter. Callers can pass in which "realm" the search should take place in, and the source can adjust how the search takes place, what attributes look like, etc.

The use case is that a software application using the Subject API has two types of users: authenticated and admin. If an authenticated user is using the application, then the Subject API should change in two respects:

1. Data should be displayed which is from the authenticated online directory, and

2. When free-form searches take place, it should only search data from the authenticated online directory.

Thus, people who for privacy reasons are not listed in the online directory might not have a name or affiliation searchable or displayed in the application. If someone is picking a person in the people picker who is not showing data in the authenticated online directory, they will need to be picked by netID or subjectID. However, admin users should be able to search anyone by name, and should be able to see the full institutional name and affiliation of any user.

Integration with Grouper 2.2

We can implement this in Grouper 2.2.

- We should add support in the LDAP adapter (we can have alternate credentials to connect)
- We should add support in the JDBC(1) adapter (we can have alternate credentials here too for connecting)
- We will have ability to define groups in grouper.properties for each realm, including a default realm. There will be some places where non admins can see admin-type realm (e.g. seeing own data)
- Otherwise API users (WS/UI) will use the realm that they are allowed

Configuring

To use this with the JDBC2 source, it is very easy. In the sources.xml, in addition to configuring the table or view, configure another table or view for another realm. There can be as many realms as you want to use, but if you use one that is not configured, you will get an exception. Also, it is a good idea to have a default realm (null), which generally should be the one that requires the fewest permissions (data is least private).

```
<init-param>
  <param-name>dbTableOrView</param-name>
  <param-value>tf_person_source_v</param-value>
</init-param>
<init-param>
  <param-name>realm__admin__dbTableOrView</param-name>
  <param-value>tf_person_source_admin_v</param-value>
</init-param>
```

Then you need to create a separate view for this realm. The tf_person_source_v would have (among other things):

- name: name from authenticated directory or netId if none
- description: name from authenticated directory if exists, netId, subjectId, if active, affiliation/orgOrDept/titleOrSchool from authenticated online directory if exists, list of other authenticed directory affiliations if exist
- search_description: anything in the description or other names the person has listed as viewable from the authenticated online directory

The tf_person_source_admin_v would have the same columns and data as the tf_person_source_v, with the following exaceptions:

- name: would be the institutional name from their primary affiliation
- description: name from primary affiliation, netId, subjectId, if active, affiliation/orgOrDept/titleOrSchool from primary affiliation, list of other affiliations
- search_description: anything from description, and any other names the person has from any affiliation or the directory

Using

There are new API methods that pass in the realm, that are backwards compatible with any source that extends BaseSourceAdapter (this should be done by anyone implementing a source). Here are the new methods:

- getSubject(String, boolean, String realm)
- getSubjectByIdentifier(String, boolean, String realm)
- getSubjectByIdOrIdentifier(String, boolean, String realm)

- getSubjectsByIdentifiers(Collection<String>, String realm)
- getSubjectsByIds(Collection<String>, String realm)
- getSubjectsByIdsOrIdentifiers(Collection<String>, String realm)
- search(String, String realm)
- searchPage(String, String realm)

So in your application, when you are calling the Subject API, just pass in the realm (in this case "admin") if you have determined that the application user is an admin. Note, if you are caching the results of the subject API, you need to take into account the realm in the cache key.