Word Problems

• Suppose that in the DAC implementation in section 2, we introduced ‘groups’ of users. How would you modify your implementation to include group access rights. Are there any similarities or differences between ‘groups’ and ’roles’ in the two implementation of the authorization module?

With the implementation of group access rights, it would be similar to a role, but not exactly. Should a user be assigned a role, that role’s access rights overwrite the user’s access rights. A specific group would have its own set of access rights, and so if a user were part of such a group, they could then have two sets of access rights by which to be authorized by once they’ve been authenticated. Should the group access rights fail to grant authorization, the system would then check the user’s authorization for the particular action.

• What other ‘constraints’ can be imposed on the RBAC implementation in section 3. Give examples and discuss briefly how you would implement them in your code.

First, only an authenticated and authorized user with the role of manager should be allowed to assign roles to other users or modify their access to certain files (or alternatively we could add another role, the administrator role, to handle this). To implement this in code, a user at all times must be the manager. If there wasn’t one in the system, an additional authentication and authorization would have to be made specifically for the assigning of this role. Secondly, no user, including the manager, should be allowed to reassign their own role. In the case of appointing a new manager, this would be have to be handled separately from how roles are currently assigned, using a new part of the program only accessible by managers.

• Another ability of RBAC-based system is to allow for ‘sessions’. Give an example of how a session can be useful for the set up in the lab, and how your code would represent the relationship between users, roles, and the given session.

Sessions would allow for a user to have multiple roles, although limited to choosing only one at a time. During a given session, such a user would only be allowed to activate a single role out of the roles that they are authorized for, and no others. To implement this in code, a user object would have to be able to hold multiple roles, and then depending on the role the user chooses for this session, they would be assigned those access rights.