Sam Chen
Breesy Reyes

CECS 378 Assignment 8 - Access Control

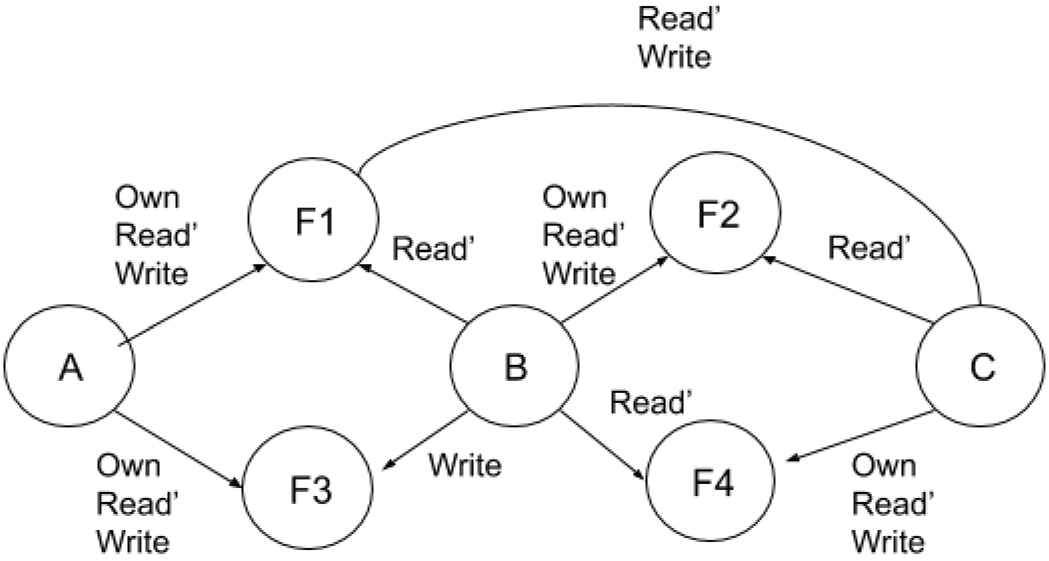
1. **Briefly describe the difference between DAC and MAC**
   For DAC, discretionary access control, the control access is based on the requestor's identity and access rule authorizations. It also permits the requestors to perform the allowed activity. For MAC, mandatory access control, it is defined based on comparing security labels with security clearances. The label will determine whether the system resource is sensitive or critical. The security clearance refers to the system entities which are eligible to access the certain resources. Since the entity may not be eligible to access the resource on its own it will tell another entity to do it, hence why it's mandatory.

2. **List and define the four types of entities in a base model RBAC system.**
   The entities are User,, an individual with the right to access the system, every user has a user ID. Role, a job function to control the system, it has authority and responsibility as a title. Permission, the approval process for a certain mode of access to an object(s). Session is a mapping between a user and a subset of roles assigned to a certain user.

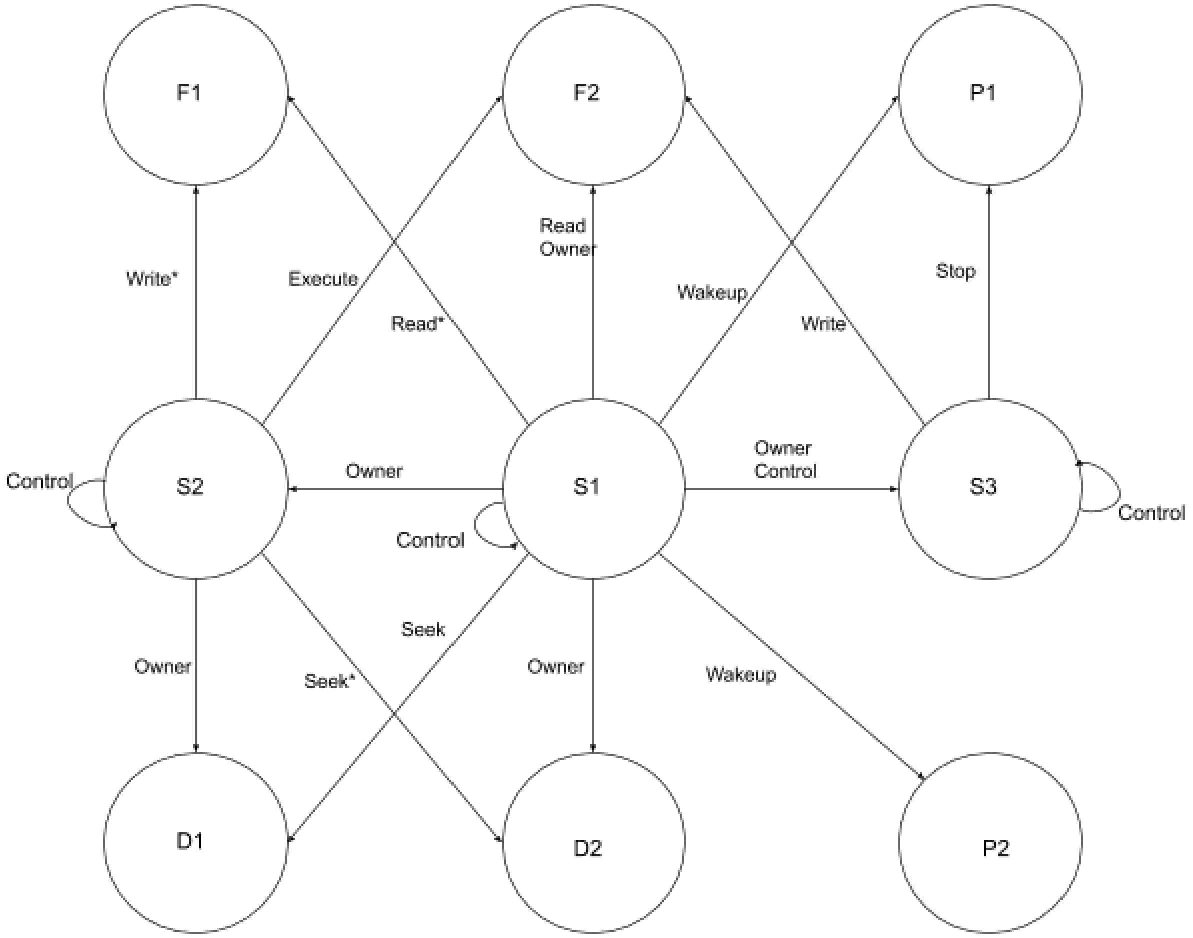
3. **In the NIST RBAC model, what is the difference between SSD and DSD?**
   In Static Separation of Duty Relations, there will be mutually exclusive roles enabled. So if one role is assigned to a user from a set, then that user cannot get any other role from the same set. It also places cardinality constraints in a set of roles. Dynamic Separation of Duty Relations means it is used to limit the permissions available to a user. It also places constraints on the roles which is then activated for a user for more permissions. Also, the constraints are defined as a pair. SSD will have constraints as a set of roles; while DSD constraints are placed as a pair.

4. **Describe three types of role hierarchy constraints**
   The three role hierarchy are mutually exclusive, cardinality, and prerequisite. Mutually exclusive, are roles in which a user has only one role in the particular set. Cardinality, are roles in which maximum numbers of users are assigned to different types of roles. For example, one or more user can belong to one role. Prerequisite, is when a user can only be assigned to a role if it is already assigned to another role.

5. **For the DAC model discussed in your textbook's Section 4.3, an alternative representation of the protection state is a directed graph. Each subject and each object in the protection states represented by a node (a single node is used for an entity that is both subject and object). A directed line from a subject to an object indicates an access right, and the label on the link defines the access right.**
   a. **Draw a directed graph that corresponds to the access matrix of Figure 4.2a.**



   b. **Draw a directed graph that corresponds to the access matrix of Figure 4.3.**



c. **Is there a one-to-one correspondence between the directed graph representation and the access matrix representation? Explain.**
Yes, there is a ono-to-one correspondence because a given access matrix generates only one directed graph, and a given directed graph yields only one access matrix.

6. **UNIX treats file directories in the same fashion as files; that is, both are defined by the same type of data structure, called an inode. As with files, directories include a nine-bit protectionstring. If care is not taken, this can create access control problems. For example, consider a file with protection mode 644 (octal) contained in a directory with protection mode 730. How might the file be compromised in this case?**
Since the file permission only has read for the group. The file has no write or execute permissions. Which means a member of the group can change the content of the file or delete the file itself. Giving permissions to the file is

useless. The file has read permission for others while the directory has no permission for others. Content of the file cannot be read by others so whoever has the permission to read is useless.

7. **In the traditional UNIX file access model, which we describe in your textbook's Section 4.4,UNIX systems provide a default setting for newly created files and directories, which the owner may later change. The default is typically full access for the owner combined with one of the following: no access for group and other, read/execute access for group and none for other, or read/execute access for both group and other. Briefly discuss the advantages and disadvantages of each of these cases, including an example of a type of organization where each would be appropriate.**
    - A default UNIX file access of full access for the owner combined with no access for group and other means that newly created files and directories will only be accessible by their owner. This means that any access for other groups or users must be explicitly granted. This is widely used by government and business.
    - A default of full access for the owner combined with read/execute access for group and none for other means newly created files and directories are accessible by all members of the owner's group. This is suitable when there is a team of people working together on a server, and in general most work is shared with the group. However there are also other groups on the server for which this does not apply. This would be used by an organization with cooperating teams may choose this.
    - A default of full access for the owner combined with read/execute access for both group and other means newly created files and directories are accessible by all users on the server. This is appropriate for organization's where users trust each other in general, and assume that their work is a shared resource. This would be used by small businesses where people need to rely on and trust each other.

8. **Consider user accounts on a system with a Web server configured to provide access to userWeb areas. In general, this uses a standard**

directory name, such as ’public_html’, in a user’s home directory. This acts as their user Web area if it exists. However, to allow the Web server to access the pages in this directory, it must have at least search (execute) access to the user’s home directory, read/execute access to the Web directory, and read access to anyWeb pages in it. Consider the interaction of this requirement with the cases you discussed for the preceding problem. What consequences does this requirement have? Note that a Web server typically executes as a special user, and in a group that is not shared with most users on the system. Are there some circumstances when running such a Web service is simply not appropriate? Explain.

In order to provide the Web server access to a user’s ‘public_html’ directory, then search (execute) access must be provided to the user’s home directory (and hence to all directories in the path to it), read/execute access to the actual Web directory, and read access to any Web pages in it, for others (since access cannot easily be granted just to the user that runs the web server). However this access also means that any user on the system (not just the web server) has this same access. Since the contents of the user’s web directory are being published on the web, local public access is not unreasonable (since they can always access the files via the web server anyway). However in order to maintain these required permissions, if the system default is one of the more restrictive (and more common) options, then the user must set suitable permissions every time a new directory or file is created in the user’s web area. Failure to do this means such directories and files are not accessible by the server, and hence cannot be accessed over the web. This is a common error. As well, the fact that at least search access is granted to the user’s home directory means that some information can be gained on its contents by other user’s, even if it is not readable, by attempting to access specific names. It also means that if the user accidentally grants too much access to a file, it may then be accessible to other users on the system. If the user’s files are sufficiently sensitive, then the risk of accidental leakage due to inappropriate permissions being set may be too serious to allow such a user to have their own web pages.

9. **Assume a system with N job positions. For job position i, the number of individual users in that position is Ui and the number of permissions required for the job position is Pi**
    a. **For a traditional DAC scheme, how many relationships between users and permissions must be defined?**
    There will only be one possible relationship
    b. **For a RBAC scheme, how many relationships between users and permissions must be defined**
    There will be U*P relationships between users and permissions

10. **In the example of your textbook's Section 4.8, use the notationRole(x). Position to denote the position associated with rolexandRole(x). Function to denote the function associated with role x.**
    a. **We defined the role hierarchy for this example as one in which one role is superior to another if its position is superior and their functions are identical. Express this relationship formally.**
    Role(x) > Role(y) ⇔ Role(x).Position > Role(y).Position ⋀ Role(x).Function = Role(y).Function
    b. **An alternative role hierarchy is one in which a role is superior to another if its function is superior, regardless of position. Express this relationship formally**
    Role(x) > Role(y) ⇔ Role(x).Position > Role(x).Function = Role(y).Function