FIT2093: Tutorial 3 sample solutions

Access Control & Security Models

Review Questions

1. Briefly define the difference between DAC and MAC.

   Discretionary access control (DAC) controls access based on the identity of the requestor and on access rules (authorizations) stating what requestors are (or are not) allowed to do. This policy is termed *discretionary* because an entity might have access rights that permit the entity, by its own volition, to enable another entity to access some resource.
   Mandatory access control (MAC) controls access based on comparing security labels (which indicate how sensitive or critical system resources are) with security clearances (which indicate system entities are eligible to access certain resources). This policy is termed *mandatory* because an entity that has clearance to access a resource may not, just by its own volition, enable another entity to access that resource.

2. List and define the three classes of subject in an access control system.

   Owner: This may be the creator of a resource, such as a file. For system resources, ownership may belong to a system administrator. For project resources, a project administrator or leader may be assigned ownership.
   Group: In addition to the privileges assigned to an owner, a named group of users may also be granted access rights, such that membership in the group is sufficient to exercise these access rights. In most schemes, a user may belong to multiple groups.
   World: The least amount of access is granted to users who are able to access the system but are not included in the categories owner and group for this resource.

3. In the context of access control, what is the difference between a subject and an object? What is an access right?

   A **subject** is an entity capable of accessing objects. Generally, the concept of subject equates with that of process. Any user or application actually gains access to an object by means of a process that represents that user or application.

   An **object** is anything to which access is controlled. Examples include files, portions of files, programs, and segments of memory.

   An **access right** describes the way in which a subject may access an object.

4. What is the difference between an access control list and a capability list?

For each object, an **access control list** lists users and their permitted access rights.
A **capability list** specifies authorized objects and operations for a user.

5. What are the three rules specified by BLP model?

   no read up: A subject can only read an object of less or equal security level. This is referred to in the literature as the simple security property (ss-property).

   no write down: A subject can only write into an object of greater or equal security level. This is referred to in the literature as the *-property.

   ds-property: An individual (or role) may grant to another individual (or role)access to a document based on the owner's discretion, constrained by the MAC rules. Thus, a subject can exercise only accesses for which it has the necessary authorization and which satisfy the MAC rules.

6. In general terms, how can MLS be implemented in an RBAC system?

   Roles can be defined by type of access and clearance level.

7. A directory is also an object to which access should be controlled. Why is it not appropriate to allow users to modify their own directories?

   The user might assign him- or herself unauthorised rights to objects to which he or she has limited legitimate access. There is also an integrity problem.

8. Why should the directory of one user not be generally accessible (for read-only access) to other users?

   The existence of certain objects should not be available to unauthorised users. For example, knowledge that Jones and Smith are working together on a project (as evidenced by shared access to many files) might be sensitive. All users should not know sometimes testing or development versions of hardware components or software systems.

Problems

1. Unix treats file directories in the same fashion as files; that is the same type of data structure, called an inode, defines both. As with files, directories include a nine-bit protection string. 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?

   Suppose that the directory **d** and the file **f** have the same owner and group and that **f** contains the text *something*. Disregarding the superuser, no one besides the owner of **f** can change its contents, because only the owner has write permission. However, anyone in the owner's group has write permission for **d**, so that any such person can remove **f** from **d** and install a different version, which for most purposes is the equivalent of being able to modify **f**.

2. In the traditional UNIX file access model, which provides a default setting for newly created files and directories, which owner may later change .

   The default is typically full access for the owner with one of the following:

   - No access for group and other
   - Read/execute access for group and none for other
   - Read/execute access for both group and other.

   Briefly discuss the advantages ad disadvantages of each of these cases, including an example of a type of organisation 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. Any access for other groups or users must be explicitly granted. This is the most common default, widely used by government and business where the assumption is that a person's work is assumed private and confidential.

   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. 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 organizations where users trust each other in general, and assume that their work is a shared resource. This used to be the default for University staff, and in some research labs. It is also often the default for small businesses where people need to rely on and trust each other.

3. The necessity of the " non read up " rule for a multilevel secure system is fairly obvious. What is the importance of the " no write down " rule?

   The purpose of the "no write down" rule, or *-property is to address the problem of Trojan horse software. With the *-property, information cannot be compromised through the use of a Trojan horse. Under this property, a program operating on behalf of one user cannot be used to pass information to any user having a lower or disjoint access class.

4. The *-property requirement for append access (blind write) {SC(S) ≤ SC(O)} is looser than for the write access {SC(S) ≥ SC(O)}. Explain the reason for this. [SC(S) refers to the security clearance level of subject, S and SC(O) is the security classification level of the object, O in a BLP model].

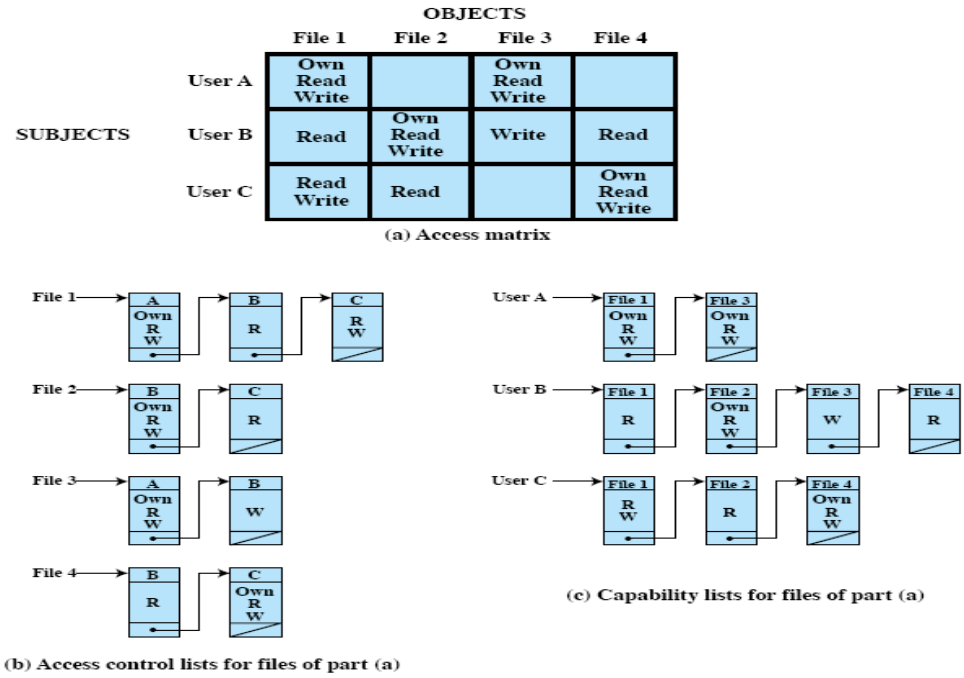
   An append function only requires the ability to update the object without observing (reading) the object. The write function requires the ability to read as well.

5. In Discretionary Access Control (DAC) the Operating System generally uses an access control matrix. Refer to the table 1 shown below. The access rights assigned to each subject are read or write. In addition the owner of the file is included in the access control matrix.
   a. Create an access control list for each of the 4 files, File 1, File 2, File 3 and File 4.
   b. Create a capability list for each of the users, User A, User B and User C.

	File 1	File 2	File 3	File4
User A	Own, read, write		Own, read, write	
User B	Read	Own, read, write	Write	read
User C	Read, write	Read		Own, read, write

Table 1: Access Control Matrix

Answer : i) Representation as per text book



		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

(b) Access control lists for files of part (a)

(c) Capability lists for files of part (a)

ii) Optional way of representation: O=Own, R=Read, W=Write

ACL FOR FILE 1	
	File 1
A	O, R, W
B	**R**
C	**R, W**

ACL for File 2	
	File 2
A	-
B	O, R, W
C	**R**

ACL for File 3	
	File 3
A	O, R, W
B	W
C	**-**

ACL for File 4	
	File 4
A	-
B	R
C	**O, R, W**

b) Capability Lists

User A	User B	User C	
O, R, W	R	R, W	File1
-	O, R, W	R	File 2
O, R, W	W	-	File 3
-	R	O, R, W	File 4

6. File access control relates largely to the secrecy dimension of security. What is the relationship between an access control matrix and the integrity of the objects to which access is being controlled?

The matrix can control mode of access (e.g., read, write, delete), of which write and delete have integrity dimensions. However, the control is very coarse: the ability to write is controlled, but what is written is not controlled. Basically, the access control matrix for integrity sets permissions to prevent unauthorized modification of objects. Controlling how subjects can read and write objects can also enforce integrity as shown in the BIBA integrity mode.

7. Consider the following three kinds of access control mechanisms:
   1. per-subject capability list (that is, one list for each subject indicates all the objects to which that subject has access)
   2. per-object access control list (that is, one list for each object indicates all the subjects who have access to that object)
   3. access control matrix

   For each kind of access control mechanism, describe the:
   a. ease of determining authorized access during execution
   b. ease of adding access for a new subject
   c. ease of deleting access by a subject
   d. ease of creating a new object to which all subjects have access by default.

   For each of the above cases, distinguish the complexity of the operation as **easy, moderately easy, moderately hard** or **very hard**.

The primary assumption for solving this question is to assume that at the point each question is asked (a,b, c,d) we have subject/object reference in hand. That is there is only ever one lookup that needs to be performed, except in the case of an access control matrix, where two lookups must be performed.

(i) **Per subject Capability List** (one list for each subject tells all the objects to which that subject has access)
(a) easy – simple O(n) searches to match the object for which access is requested.
(b) easy – append or insert new object into subject's list (again O(n))
(c) moderately easy – again, assuming the subject is known
(d) very hard – each subject must be updated individually. In per-subject ACL the difficulty is with object accesses that may span multiple subjects. Either the access right has to appear in each subject's list, or a means of linking lists has to be provided.
(ii) **Per Object ACL**(one list for each object tells all the subjects who have access to that object)
(a) easy – simple O(n) search to match the subject requesting access
(b) moderately hard – append or insert new subject into each object's list is hard. Per-subject controls are difficult to implement in a per-object environment.
(c) easy – simple O(n) search and removal
(d) easy – simple creation of new objects and insertion of all subjects to that list – O(n)
(iii) **Access Control Matrix** (the matrix is the combination of per-subject (e.g., rows) and per-object (e.g., columns) structure. It facilitates all operations listed in the question. The disadvantage to the matrix is that it is sparse)
(a) Moderately easy – search m subjects, n objects for lookup – O(mn)
(b) moderately easy – again add the new subject's row , add the accesses for the all n objects of the new subject – O(mn)
(c) moderately easy – again search m subjects, n objects for lookup and delete/alter the accesses – O(mn)
(d) easy – create column, then cycle through m subjects – O(m)

8. Consider three users in a Unix system **Srini, Campbell** and **Maria** such that **Srini** and **Campbell** are in the same group **staff**, but **Maria** does not belong to this group. Consider the following files, with different access information.

```
- --- --- ---  1  srini  staff  4257 Sept 5 14:00 file01
- --- --- rw-  1  srini  staff  4257 Sept 5 14:00 file02
- --- rw- ---  1  srini  staff  4257 Sept 5 14:00 file03
- --- rw- rw-  1  srini  staff  4257 Sept 5 14:00 file04
- rw- --- ---  1  srini  staff  4257 Sept 5 14:00 file05
```

```
- rw- --- rw-  1  srini  staff  4257 Sept 5 14:00 file06
- rw- rw- ---  1  srini  staff  4257 Sept 5 14:00 file07
- rw- rw- rw-  1  srini  staff  4257 Sept 5 14:00 file08
```

Fill in the following table based on the accessibility of each file by each one of the user (by putting yes or no in the box).

This question needs the understanding of Unix file permissions. The first thing one needs to understand is that the permissions are read from left to right (**owner, group, world)**

	Srini		Campbell		Maria	
	reads	writes	reads	writes	reads	writes
file01	No	No	No	No	No	No
file02	No	No	No	No	Yes	Yes
file03	No	No	Yes	Yes	No	No
file04	No	No	Yes	Yes	Yes	Yes
file05	Yes	Yes	No	No	No	No
file06	Yes	Yes	No	No	Yes	Yes
file07	Yes	Yes	Yes	Yes	No	No
file08	Yes	Yes	Yes	Yes	Yes	Yes

Table 2: Access Rights

Remember that the permissions are short circuited in a sense, because as soon as the system knows you are granted or denied access, it discontinues the checking process.