Homework 6 (Total Points: 100), Due Date*: 12:00pm (noon) 10/30/2019, Cutoff Date*: 12:00pm 11/01/2019

***Late penalty will apply for past-due late submission **Submission will NOT be accepted after the cutoff deadline**

Submission: handwritten hardcopy at the beginning at the class (Email to wzhu1@msudenver.edu muse be used for late submission and the submission time is the moment when the email arrives at the instructor's inbox. Any copy in a student's mailbox cannot be used as a proof of submission.)

Part I. Review Questions for reading the textbook (No submission; however, the relevant contents will be included in Exams as True/False, Multiple-Choices, and/or Filling-in-the-Blanks questions for up to 20% of the total points.)

- **Textbook, Page 133-134, Review Questions**
  4.1 Briefly define the difference between DAC and MAC.
  4.2 How does RBAC relate to DAC and MAC?
  4.3 List and define the three classes of subject in an access control system.
  4.4 In the context of access control, what is the difference between a subject and an object?
  4.5 What is an access right?
  4.6 What is the difference between an access control list and a capability ticket?
  4.7 What is a protection domain?
- **Textbook, p135, Problem 4.6**

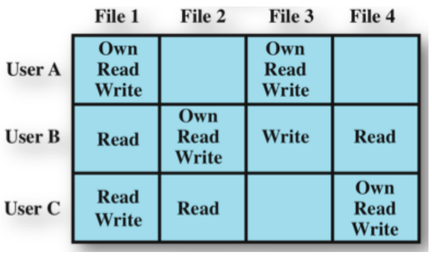
Part II. Problems (Submission Required Please!)

Problem A. List the authentication security issue for each one of the following definitions.

a) Adversary repeats a previously captured user response.
b) It is directed at the user file at the host where passwords, token passcodes, or biometric templates are stored.
c) Adversary attempts to learn the password by some sort of attack that involves the physical proximity of user and adversary.
d) Adversary attempts to achieve user authentication without access to the remote host or the intervening communications path.
e) Adversary attempts to disable a user authentication service by flooding the service with numerous authentication attempts.
f) An application or physical device masquerades as an authentic application or device for the purpose of capturing a user password, passcode, or biometric

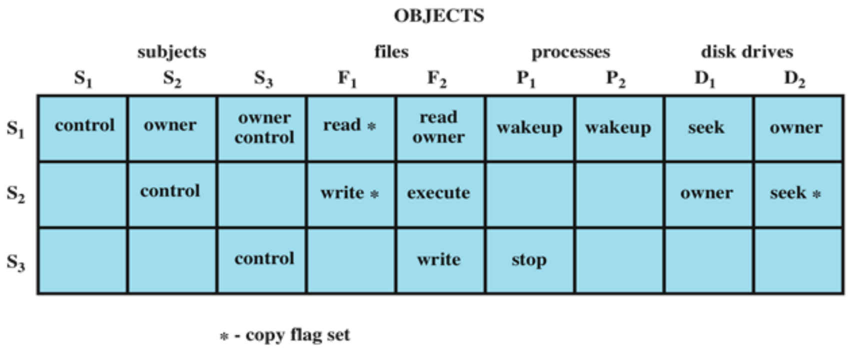
Textbook, P134, Problem 4.1. For the DAC model discussed in Section 4.3, an alternative representation of the pro-tection state is a directed graph. Each subject and each object in the protection state is represented by a node (a single node is used for an entity that is both subject and object). A directed line from a subject to an object indicates all the access rights by this subject to this object, and the label on the link defines all these access rights.

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.



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

Figure 4.2a



OBJECTS

	subjects			files		processes		disk drives	
	S_1	S_2	S_3	F_1	F_2	P_1	P_2	D_1	D_2
S_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
S_2		control		write *	execute			owner	seek *
S_3			control		write	stop			

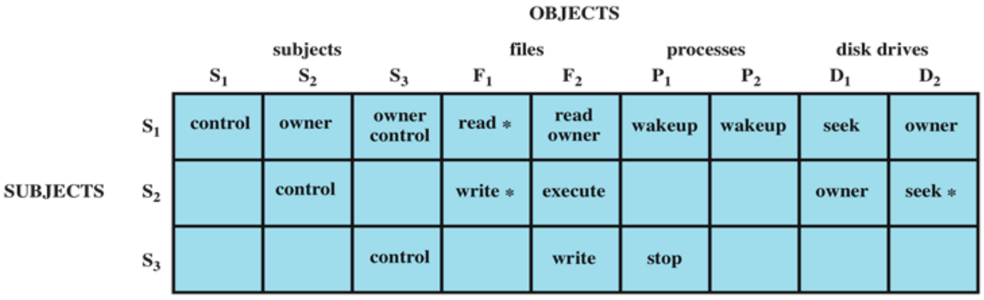
SUBJECTS

* - copy flag set

Figure 4.3

(There are more problems on the next page!)

Problem B. Using the given access matrix on the right, for each one of the following messages generated by the operating system or an access control interface module in response to an access attempt, describe (1) ***which controller*** (access control module) receives and processes this message, (2) ***whether*** this access attempt is ***authorized*** or ***rejected***, and (3) ***how*** the access matrix is ***updated*** and what is *w* if any. (Hint: each attempt starts with the initially given access matrix; you just need plot the ***modified*** *part* or state "***no change***". e.g., "A[S2, P2]=wakeup" is not enough, please plot the entry labeled with S2/P2 as the row/column indices and fill in *wakeup*.)



		OBJECTS								
			subjects		files		processes		disk drives	
		S_1	S_2	S_3	F_1	F_2	P_1	P_2	D_1	D_2
	S_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
SUBJECTS	S_2		control		write *	execute			owner	seek *
	S_3			control		write	stop			

* - copy flag set

a) S1, *read*, F1 b) S2, *stop*, P2 c) S3, *wakeup*, P1 d) S3, **transfer** *write* to S1, F2
e) S2, **grant** *seek** to S3, D1 f) S1, **delete** *write* from S3, F2 g) S1, w ← **read** S2, D1 h) S2, **create object** D3
i) S1, **destroy object** D2 j) S3, **create subject** S4 k) S1, **destroy subject** S2

Textbook, p134-135, Problem 4.5 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 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? (Hint: How can anyone in the owner's group modify the contents of this file?)

Problem C. On a UNIX system, consider ***executable file A*** with userID = 88, groupID=123, protection mode=751, StickyBit=0, ***text file B*** with userID = 88, groupID=123, protection mode=640, SetUID=0, SetGID=0, StickyBit=0, ***user X*** with userID=88 and groupID=123, ***user Y*** with userID=99 and groupID=123, and ***user Z*** with userID=77 and groupID=456. ***Procedure R*** in ***file A*** reads ***text file B*** and ***Procedure W*** in ***file A*** writes to ***text file B***. (Hint: the protection mode in Octal representing the 9 permission bits.)

a) Using a text editor with SetUID=0, SetGID=0, which user(s) can read ***text file B***?
b) Using a text editor with SetUID=0, SetGID=0, which user(s) can write to ***text file B***?
c) If for ***file A***, SetUID=1, SetGID=0, which user(s) can execute ***file A*** to read ***text file B***?
d) If for ***file A***, SetUID=1, SetGID=0, which user(s) can execute ***file A*** to write to ***text file B***?
e) If for ***file A***, SetUID=0, SetGID=1, which user(s) can execute ***file A*** to read ***text file B***?
f) If for ***file A***, SetUID=0, SetGID=1, which user(s) can execute ***file A*** to write to ***text file B***?

Problem D. On a UNIX system, consider ***directory A*** with userID =12, groupID=333, protection mode=731, ***text file B*** with userID 67, groupID=555, protection mode=700, SetUID=0, SetGID=0, StickyBit=0, ***user X*** with userID=12 and groupID=555, ***user Y*** with userID=67 and groupID=333, and ***user Z*** with userID=34 and groupID=999. (Hint: the protection mode means the same above.)

a) Which user(s) can list files in ***directory A***?
b) If for ***directory A***, StickyBit = 0, which user(s) can create/rename/delete files in ***directory A***?
c) If for ***directory A***, StickyBit = 1, which user(s) can create files in ***directory A***? Which user(s) can rename/delete ***text file B*** that is assumed to be in ***directory A***?
d) Which user(s) can descend into ***directory A*** or search it for ***text file B*** that is assumed to be in ***directory A***?
e) If for ***directory A***, SetGID = 0 and protection mode is changed into 777, what are the groupID and userID of a new file created by ***user Z*** in ***directory A***?
f) If for ***directory A***, SetGID = 1 and protection mode is changed into 777, what are the groupID and userID of a new file created by ***user Z*** in ***directory A***?