Joshua Main-Smith
September 27, 2020

Module 4 Assignment

1. **(10 points) Read about UNIX access control in section 4.4 of your book. 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?**

   This is a little easier to see in binary than in decimal:

Permission Group	Decimal	Binary
Owner	6	0110
Group	4	0100
All	4	0100

For each bit, from left to right, access control is defined as (1) special permission flag, (2) read permission, (3) write permission, and (4) execute permission. So, according to the above table, the owner of the file has read and write permission, the group and all others only have read permission. The table below shows the protection mode for the file in question:

Permission Group	Decimal	Binary
Owner	7	0111
Group	3	0011
All	0	0000

The owner of the directory has read, write and execute permissions, the group has write and execute permissions, and all others have no permission. A potential problem with this setup is with the group having write access gives anyone within this group the ability to create/rename/delete files in the directory. Someone in the group could delete the file above, create a similar one and would have ownership permission of the file. So, it would be pertinent in assigning trusted individuals to the group. Another issue I see is the All group has read access to the file but has no access to the directory. So, unless there is another way the All group can access the file, they wouldn't have the ability to exercise the permission that they've been given.

2. **(10 points) Assume a system with N job positions. For job position i, the number of individual users in that position is U_i. and the number of permissions required for the job position is P_i.**

    a. **For a traditional DAC scheme, how many relationships between users and permissions must be defined?**

    The relationship between users and permissions can be calculated from the number of objects by the number of users, i.e. NxU=P. Each new user/object needs to be added to the access control matrix, increasing the number of user-permission relations.

    b. **For a RBAC scheme, how many relationships between users and permissions must be defined?**

    Unlike DAC above, the role-based scheme a user is only related to a permission via their role. If those enrolled in Role 1 have three permissions to access some object, Alice would be granted those three permissions once enrolled in Role 1. So, the number of user-permission relations would be calculated by the number of roles (R_i) by the number of objects (N), or R_ixN.

3. **(10 points) Consider a computer system with three users: Alice, Bob, and Cyndy. Alice owns the file alicerc, and Bob and Cyndy can read it. Cyndy can read and write the file bobrc, which Bob owns, but Alice can only read it. Only Cyndy can read and write the file cyndyrc, which she owns. Assume that the owner of each of these files can execute it.**

    a. **Create the corresponding access control matrix.**

    r = read, w = write, x = execute.

	alicerc	bobrc	cyndyrc
Alice	x	r	
Bob	r	x	
Cyndy	r	rw	rwx

b. Cyndy gives Alice permission to read cyndyrc, and Alice removes Bob's ability to read alicerc. Show the new access control matrix.

	alicerc	bobrc	cyndyrc
Alice	x	r	r
Bob		x	
Cyndy	r	rw	rwx

4. (20 points) Consider a hypothetical enterprise. It can be the same organization you thought about in the Module 1 Discussion. List some subjects (e.g. people/roles) and objects (e.g. files, software applications). What are the access requirements? Would you want mandatory access control or discretionary access control or role based access control?

I chose a metropolitan police department with a forensics lab, detective wing, and a hierarchy of roles. The permissions granted are based off of a particular role's minimal necessity in relation to the object. For instance, the lead forensic investigator doesn't need access to the arms/munitions cabinet to function effectively in the role, but would need access in seeing, using, and handling the forensics test kits. An officer doesn't need write access for case files, but they would need access to what it says in order to prepare for a court hearing. The higher ranking officials, such as Lieutenant, Captain, etc. don't need access to writing or handing reports, or to the evidence, but may need read access in the event that they are also called to a court hearing, or for public relations purposes. Since there are many people in the same role, a role-based approach seemed more appropriate in assigning permissions to particular individuals.

Roles	Case Files	Evidence Reports	Physical Evidence	Digital Evidence	Forensics Test Kits	Arms / Munitions
Forensics Lab Tech	r	rw	rx	rx	rx	

Lead Forensics Investigator	r	rwx	rx	rx	rwx	
Detective 1	rw	r	r	r		rwx
Detective 2	rw	r	r	r		rwx
Officer 1	r	r				rx
Officer 2	r	r				rx
Sergeant	rx	rx				rwx
Lieutenant	r	r				
Captain	r	r				
Chief	r	r				

Name	Role
Jimmy	Officer 1
Jeffery	Officer 2
Johnny	Sergeant
Jackie	Lieutenant
Jenny	Captain
Josephine	Forensics Lab Tech
Joseph	Lead Forensics Investigator

Joshua Main-Smith
September 27, 2020

Jack	Detective 1
Joe	Detective 2
James	Chief
Jerome	Officer 1
Jake	Officer 2
Jen	Detective 1
Jasmine	Detective 2
Josh	Forensics Lab Tech