Vincent Yasi
CS 370 -Intro to Security-

Problem Set 6

Access Control Concepts

Q1 [2 pts]: What is the difference between a "role" in RBAC and a "group" commonly used in UNIX?

A group is simply a collection (group) of users on a system. They are not necessarily related or share any common features.

A role, however, denotes a purpose or common characteristic. A role itself often has certain defining features and overall represents a certain type of person or position. Therefore, people who share the same role share the same characteristics (unlike with a group in Unix).

Q2 [3 pts]: What is separation-of-duty? And what is the difference between static separation-of-duty (SSD) and dynamic separation-of-duty (DSD)

Separation of duty means to divide up permissions and similar things based on the duties or needs of a user. It limits the permissions to those needed by the user, as opposed to just giving them full permissions.

Static separation of duties means a user's permissions are assigned beforehand, based on role, etc. It is the typical way permissions are thought of, where permissions are assigned and those are "your permissions".

Dynamic separation of duties instead assigns permissions on a session by session basis, often based on criteria or the situation at the time. For each session, the user is dynamically assigned their permissions, and they can change if the need arise.

Role-Based Access Control

Q3 [8 pts]: Consider the following scenario. An organization employs product managers, programmers and testers. The organization operates with the following kinds of files: development code and executables, testing code and executables, test reports, and production code and executables.

> **Product Managers can view and execute the development executables and production executables to verify correctness. Programmers can create, edit, delete, and execute development code and executables.**
>
> **Programmers can also promote development code to the test level.**
>
> **Testers can edit, delete, and execute test code and executables. The testers write test reports that can be read by everyone. The testers can promote test code to production level or demote it back to**

development.

Everyone can view and execute production code and executables.

Eve is the product manager, Alice and Bob are programmers. Carol and Dave are testers

Would the access control for the scenario above benefit from being implemented in a RBAC system? If yes, explain why and create access matrices that define an RBAC that would enforce this scenario? If not, describe why not and present another scenario that would be better defined as an RBAC system rather than a straight DAC.

This scenario would benefit from an RBAC system, as there are clearly defined roles, and each role has clearly defined permissions associated with them. Therefore, assigning permissions to roles, and then roles to people, would work well. Each person above has a specific role, and so each can be assigned their role, as opposed to individual permissions.

	Development Code	**Development Executable**	**Test Code**	**Test Executable**	**Test Report**	**Production Code**	**Production Executable**
Manager	---	Read, Execute	---	---	Read	Read, Execute	Read, Execute
Programmer	Create, Edit, Delete, Execute, Promote	Create, Edit, Delete, Execute	---	---	Read	Read, Execute	Read, Execute
Tester	---	---	Edit, Delete, Execute, Promote, Demote	Edit, Delete, Execute	Create, Edit, Read	Read, Execute	Read, Execute

Q4 [7 pts]: A company has 20 job functions. On average there are 200 employees in each job function. Similarly, on average an employee in each job function needs 1500 permissions to properly execute their task. Compare the number of assignments that need to be managed i) when using a DAC model vs. ii) when using RBAC model. Generalize the comparison to when the number of job functions is N, number of employees per job function is $U_{i,}$ where i indexes the job-function, and the number of permissions required per job function is P_i

For a DAC model, each person needs to be assigned permissions on an individual basis. If there are N job functions, and U employees per job, that makes U*N individuals who need individualized permission assignments. Then, for each job function there are P permissions needed. P permissions need to be assigned for each person, which gives an overall number of permission assignments of P*U*N, which can get very large, very quickly, and make much work for the system administrators. For the example above, this would be 1500*200*20, or 6,000,000 individual assignments.

For an RBAC model, permission sets can be made per job, and then each job assigned to the particular individual. This greatly cuts down on permission assignments for administrators. Using the generalization above, each job N would have P permissions, giving N*P permissions needed to be assigned. However, once they are assigned, the roles can be given to each individual without having to assign the permissions again. Therefore, with the numbers above, the number of permission assignments is only 20*1500, or 30,000, much lower than DAC. Then, for each person that needs that role, it is only one assignment per person (if they have only one role), instead of 1500 each time.

Mandatory Access Control Models

Q5 [4 pts]: What is *-property in BLP confidentiality model and why is it needed?

This property is that one can only write to a file of the same security level or higher. This is needed because otherwise higher level material could be written to a lower security level, and the BLP model aims to protect secrecy levels by allowing material to only become more secure, not less.

Q6 [4 pts]: Compare and contrast BLP and Biba models.

BLP is concerned with the security of files, etc. It aims to increase the security level of a file as it increases in level, with the most secure at top and least at bottom. It does this by controlling information flow only from less to more secure, and protecting higher levels from lower level clearances.

Biba, meanwhile, is concerned with protecting the integrity of files, etc. It wants the most trustworthy files to be at the top, and least trustworthy at bottom. It does this by controlling data flow from most trustworthy down only, and allowing running of programs from the same or lower levels of integrity.

Both systems aim to control data and files, but each has a different end goal (security vs integrity). They want to protect files of "higher attribute" from those of lower, which may "spoil or degrade" them. Which system to use would be based on what attribute you want to protect.

Q7 [2 pts]: What is the difference between a security level and an integrity level?

As touched upon above, a security level is a measure of how secure or secret a thing is. It can also be a measure of how detrimental the information contained within can be if known by unwanted parties. More secure knowledge does not want to be allowed to move to lower levels.

Integrity, on the other hand, is a measure of how trusted a thing is, or how certain one is that it will perform how it is expected to. Again, more trusted material wants to be kept from less trusted, as this can compromise the integrity of the material and bring its reliability into question.

Q8 [3 pts]: How is Chinese Wall model different from BLP and Biba?

Chinese Wall, meanwhile, is concerned with conflicts of interest and keeping material that may conflict separate. It allows access to material only if a user's previous actions are not in conflict with that material.

As opposed to BLP or Biba, Chinese Wall does not work on a layered/level system, but instead works with a group one. It categorizes material into certain datasets/groups, and these datasets are then grouped into larger Conflict Groups based on if the datasets pose a possible conflict of interest. By controlling which datasets can be accessed in a Conflict Group, it controls the material which can be accessed, and therefore protects against conflicts of interest. BLP and Biba, meanwhile, use which level material is at to control it, instead.

Q9 [6 pts]: When using DAC under MAC in BLP:

a) **Does a user get access to an object if MAC policy doesn't permit it? Explain why or why not.**

A user does not, as MAC provides the overall top layer permissions for the system. Permissions can be modified within an MAC layer using DAC, but the overall control and structure of MAC still stands to provide the protections MAC is meant to give. If these were bypassed, all the protection of MAC would be compromised.

b) **Does a user get access to an object if DAC policy doesn't permit it? Why or why not.**

A user does not get access here, either. The DAC policies still apply here. Much like with MAC, the DAC protections only remain if the DAC policies are followed. If they are broken, even if MAC still stands, the protections that are enforced by the DAC policy are gone. With a DAC under MAC system, both rulesets are in affect, and both apply.

Q10 [8 pts]: The table below lists subjects, objects, and their associated security levels. The relationship between the levels is as follows: purple > green > orange

Subject	Subject Clearance	Object	Object Classification
Alice	Green	Yoyo	Purple
Bob	Purple	XRay	Green
Carol	Orange	Zebra	Green

a) Compute whether the specified subject has read or append (i.e., write but not necessarily read) access to the specified object (see table below) following the Bell LaPadula model.

Subject	Object	Rights
Alice	XRay	Read and Append, as same level
Bob	Zebra	Read, but no Append, as file is lower level
Carol	Yoyo	Append, but no Read, as file is higher level
Carol	Zebra	Append, but no Read, as file is higher level

b) The security labels are updated to include project categories, p1, p2, and p3. The updated labels are shown in the table below. Re-evaluate the rights (read or append) associated with each subject and object pair following the Bell LaPadula model.

Subject	Subject Clearance	Object	Object Classification
Alice	Green:{p1,p2}	Yoyo	Purple:{p1}
Bob	Purple:{p2}	XRay	Green:{p1, p2}
Carol	Orange: {p1, p3}	Zebra	Green: {p3}

Subject	Object	Rights
Alice	XRay	Read and Append, as same level and same projects
Bob	Zebra	None, as while higher level than file, file is of another project Bob does not have permissions for
Carol	Yoyo	Append, but no Read, as file is higher level and project 1
Carol	Zebra	Append, but no Read, as file is higher level and project 3

Q11 [8 pts]: The table below lists subjects, objects, and their associated ***integrity*** levels. The relationship between the levels is as follows: purple > green > orange

Subject	Subject Level	Object	Object Level
Alice	Green	Yoyo	Purple
Bob	Purple	XRay	Green
Carol	Orange	Zebra	Green

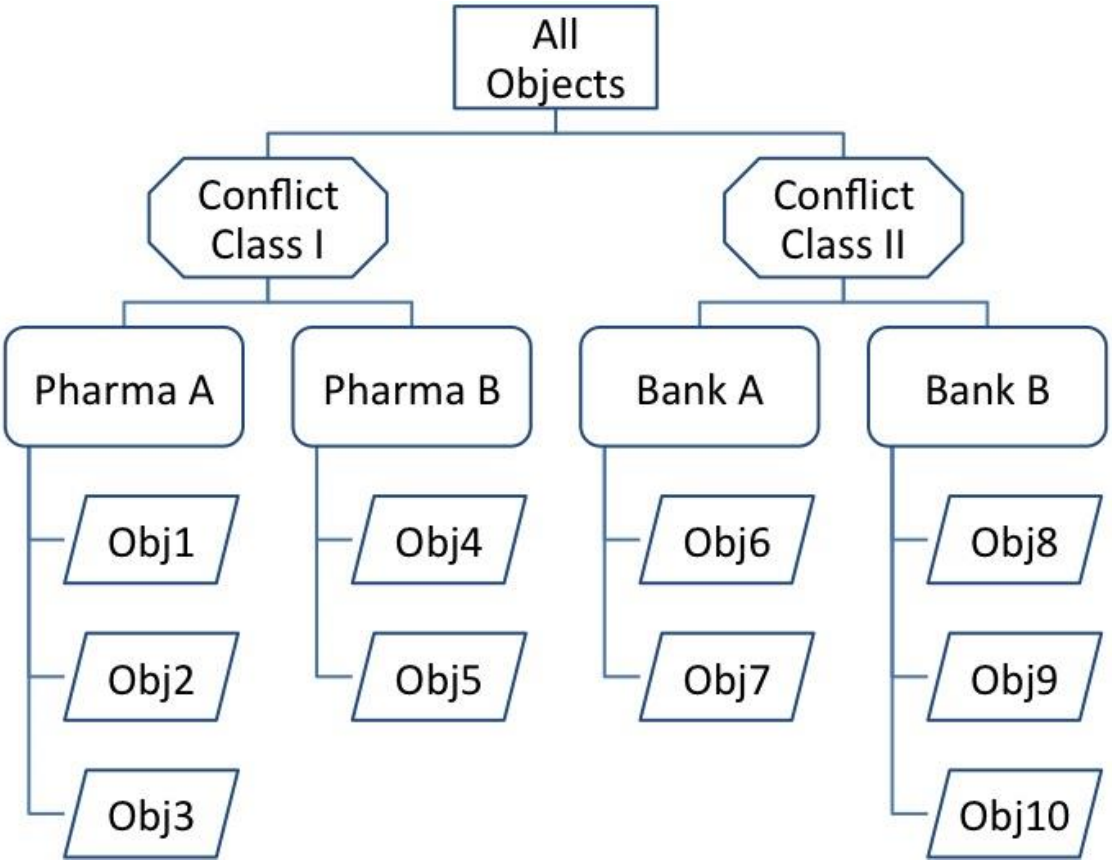
b) **Compute whether the specified subject has *observe (read)* or *modify (append or update)* access to the specified object (see table below) following the *Biba Strict Integrity Policy*.**

Subject	Object	Rights
Alice	XRay	Observe and Modify, as same level
Bob	Zebra	Modify, but no Read, as file lower level
Carol	Yoyo	Read, but no Modify, as file is higher level
Carol	Zebra	Read, but no Modify, as file is higher level

c) **The *integrity* labels are updated to include project categories, p1, p2, and p3. The updated labels are shown in the table below. Re-evaluate the rights (modify or observe) associated with each subject and object pair following the Biba model.**

Subject	Subject Class	Object	Object Class
Alice	Green:{p1,p2}	Yoyo	Purple:{p1}
Bob	Purple:{p2}	XRay	Green:{p1, p2}
Carol	Orange: {p1, p3}	Zebra	Green: {p3}

Subject	Object	Rights
Alice	XRay	Observe and Modify, as same level and projects
Bob	Zebra	None, as higher level, but different project
Carol	Yoyo	Read, but no Modify, as file is higher level and same project
Carol	Zebra	Read, but no Modify, as file is higher level and same project



Q12 [5 pts]: Figure above depicts organization of objects into datasets (e.g., Bank A) and conflict of interest classes (e.g., Conflict Class I) at consulting firm ConFirm X that uses Chinese Wall access model. Jane, Bob, Emily, Marcus, and Alice are consultants with the firm. Assume that the consultants currently have no other accesses than those explicitly stated. Please answer the following with respect to the above figure when using a Chinese Wall access model.

a) **Can Bob be allowed to read Obj 6 and Obj2? Explain why or why not.**

Yes, as they are in two different Conflict Classes, and so there is no conflict.

b) **Can Jane be allowed to read Obj7 and Obj10? Explain why or why not.**

No, as they are both in the same Conflict Class, but different datasets, and so they conflict.

c) **Can Emily be allowed to read Obj1 and write to Obj9? Explain why or why not.**

No, as once you have read an object, you can only write to objects in the same dataset to protect against possible secondhand conflicts if someone else reads Object 9.

d) **Can Marcus be given read and write access to Obj8 and write access to Obj10? Explain why or why not.**

Yes, as these are in the same dataset, so no conflicts will occur from just those objects.

e) **Can Alice be given read and write access to Obj6 and Obj 3? Explain why or why not.**

No, as these are in different datasets, for the same reasons as c).