**Hsiang Lo**

**CS 370 Introduction to Security Week 6: Problem Set 6**

Instructor Name: Rakesh Bobba

# Introduction / Purpose

The purpose of this assignment is to help you gain a better understanding and insight into role-based and mandatory access control models covered in Week 6.

Before beginning make sure you have watched the lecture videos on the following and completed the associated practice quizzes.

* Introduction to Role-based Access Control
* Role-Based Access Control Models
* Role Engineering
* Introduction to Mandatory Access Control, Bell-LaPadula (BLP) Model
* Biba Integrity Model
* Chinese Wall Model

Please make sure you read Chapter 8 and Chapter 9 till 9.2.2 from text book

# Instructions/Questions

Please answer the questions below.

## Access Control Concepts

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

A role in RBAC refers to permissions and is defined by a set of permissions and a group in UNIX is defined as a collection of permissions, and possibly other roles.

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 is the concept of having more than one person charged with the completion of a task and it’s also referred to as segregation of duty. The concept of Separation-of-duty stems from the prevention of conflict of interests.

The difference between static separation-of-duty (SSD) and dynamic separation-of-duty (DSD) differs in that Static separation-of-duty focuses on cardinality constraints on a set of roles. While Dynamic Separation of Duty focuses on prevention of role activation in a given session or time.

## 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.

Given the scenario, it would definitely be beneficial to implement the provided data and information in a RBAC system. This is because there are clearly defined roles that can be further broken down into specific tasks that are performed by objects. And given that different tasks are provided to different subjects of different roles, it can be assumed that it’s beneficial to build and construct a RBAC model.

User-to-Role Mapping

|  |  |  |  |
| --- | --- | --- | --- |
|  | Product Manager | Programmer | Tester |
| Eve | X |  |  |
| Alice |  | X |  |
| Bob |  | X |  |
| Carol |  |  | X |
| Dave |  |  | X |

Role-to-Permission Mapping

|  |  |  |  |
| --- | --- | --- | --- |
|  | Project Manager | Programmer | Tester |
| Test Code |  |  | Edit, Delete, Execute, Promote, Demote |
| Test Executable |  |  | Edit, Delete |
| Product Code | View, Execute | View, Execute | View, Execute |
| Product Executable | View, Execute | View, Execute | View, Execute |
| Development Code |  | Create, Edit, Delete, Execute |  |
| Development Executable | View, Execute | Execute, Create, Edit, Delete |  |
| Report |  |  | Write |

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 ˜on is Ui, where i indexes the job-function, and the number of permissions required per job function is Pi

Given that there are 200 employees with 20 job functions where each function requires up to 1500 permissions to properly execute their task. Using DAC model, which require the user to assign permissions to each user by hand as DAC require user-role assignment. This would result in 20 \* 200 \* 1500 = 6,000,000 assignments with formula Ui \* Pi \* N.

On the other hand, for RBAC model, it would simply be 20\*1500 + 200\*20 = 34000 assignments. This is because for a given job function, you need to assign a permission for it manually once, or 20 \* 1500 times. And then you need to assign 20 people with 20 job functions, so 30000 + 200 \* 20 which is equal to 34000. The formula for this is equal to Ui \* Pi + N \* Ui

## Mandatory Access Control Models

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

In BLP confidentiality model, the \*-Property is the 2nd property which contributes to no write down. It basically says that a subject can only append or write-only into an object of greater or equals security level. Also a subject can only read+write into an object of ame security level.

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

The main difference between BLP and Biba Model is in that BLP focuses on security while Biba focuses on Integrity. BLP model have No Read Up in that subject can only read an object of less or equal security level and No Write Down in that a subject can only append (write-only) into an object of greater or equal security level. While BIBA model states that No Write Up and No Read Down in that subject can write to an object only if its higher level and that subject can read to only if subject is than or equal to object’s level.

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

Security level deals with access decisions and the prevention of revealing information. while integrity deals with confidence that a program will execute correctly or that data is accurate and/or reliable. The work in Dual as stated from Q6.

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

Chinese Wall Model differs from BL and Biba in that it deals with or addresses conflict of interest policies. Chinese Wall Model also structure their program into information categories of different level. The way of accessing Chinese Wall Model is vastly different from BL and Biba which are purely based on levels but Chinese Wall Model is based on grouping.

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

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

No, using DAC under MAC in BLP would not work as you need both DAC and MAC in order for this to work/gain access.

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

No, using DAC under MAC in BLP would not work as you need both DAC and MAC in order for this to work/gain access.

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 |

1. 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.

Under Bell LaPadula or BLP model, in order to read/append but not necessarily read access to the specified objects, you must follow the Simple Security Property which translates to No Read Up. Subject can only read an object of less or equal security level. Alice Can read/append to XRay and Zebra but not Yoyo. Bob may read/append to Yoyo, XRay and Zebra. And Carol cannot read/append to any of them.

|  |  |  |
| --- | --- | --- |
| Subject | Object | Rights |
| Alice | XRay |  |
| Bob | Zebra |  |
| Carol | Yoyo |  |
| Carol | Zebra |  |

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/Append |
| Bob | Zebra | Read |
| Carol | Yoyo | Append |
| Carol | Zebra | Append |

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 |

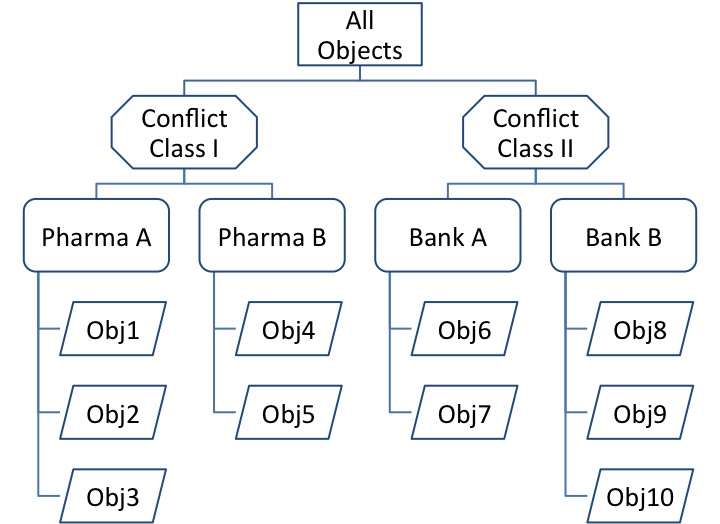
1. Compute whether the specified subject has ***observe (read)*** or ***modify (append or update)*** access to the specified object (see table below) following the ***Biba Strict Integrity Policy***.

|  |  |  |
| --- | --- | --- |
| Subject | Object | Rights |
| Alice | XRay | Observe and Modify |
| Bob | Zebra | Modify |
| Carol | Yoyo | Observe |
| Carol | Zebra | Observe |

1. 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 | Modify and Observe |
| Bob | Zebra | Modify |
| Carol | Yoyo | Modify |
| Carol | Zebra | Observe |



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.

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

No because Bob can only access the same Dataset (DS).

1. Can Jane be allowed to read Obj7 and Obj10? Explain why or why not.

No because they are in the same conflict of Interest class (CI).

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

Yes because they are not in the same Dataset (DS) and not the same Conflict of Interest class (CI).

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

Yes because they are not only in the sane Conflict of Interest class (CI) but also in the same Dataset (DS).

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

No Because the two object needs to be in the dataset (DS) in order to receive for read and write access.

# Submission Details

Submit a PDF file with the questions and your corresponding answers

The assignment is worth 60 points. It is due Wednesday of Week 7 at Midnight.