**Submission Template:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | | **Sadam Hussain Ganie** | | | |
| **Roll No.** | | **23CSM2R20** | | | |
| **Subjects** | | Admin, Customer, Manager, Employee | | | |
| **Objects** | | Admin Data, PredictionData, Manager Data, Employee Data, Manager Salary, Employee Salary. | | | |
| **Rights** | | Read, Edit. | | | |
| **S. No.** | **Services provided** | | | **Model used** | **Reason** |
| **1.** | **Employee** | | Read access to sensitive data | Bell-LaPadula | To maintain the confidentiality of the data, the no-read up property of Bell-Lapadula model is helpful. Employees should not be able to read the data of the manager, or admin, or customer. |
| Edit access to personal data | Biba Model | Employees are given access to read and edit the personal data. For read operation, Bell-Lepadula is used. But for ensuring edit access, Biba Integrity model is used. Using the “no-write up” property of Biba Model, it can be ensured that an employee is able to edit his personal details only. Employee is unable to edit his ‘salary’, this is ensured by assigning higher classification to employee salary. Refer Model explanation at the end for more clarification. |
| Restriction to modification of sensitive data | Combination of Bell-Lepadula and Biba model.  Bell-Lepadula preserves confidentiality whereas the Biba model preserves integrity. | Both Bell-Lepadula and Biba models are used to preserve the confidentiality and integrity of other data. Biba model restricts the employee from editing sensitive information, and Bell-Lepadula restricts the employees from reading the sensitive information of higher classified objects like employees can’t read admin’s data. |
| **2.** | **Administrator** | | Full control over data | Bell-Lepadula, & Biba Model. | As admin can read or write any data except prediction data, this can be achieved using a combination of Biba and Bell-Lapadula model. As admin is assigned a higher classification, it can read/edit all the data and this can be easily achieved using Bell-Lapadula and Biba Model. |
|  |  | | No modification to prediction data | Bell-Lepadula, Biba Model | We don’t want admin to edit the prediction data but prediction data is assigned lower classification than admin. In this case, we will assign a different category to ‘prediction data’ as that of ‘admin’ category. By assigning different categories, we can make prediction data uneditable. |
| **3.** | **Customer** | | View only access to Prediction data | Bell-Lapadula Model | We need to give read access to customers on ‘prediction data’. This can be achieved using the ‘no-read up’ property of Bell-Lapadula model. By assigning the same classification to both ‘customer’ and ‘prediction data’, customers can read ‘prediction data’. Classification of ‘prediction data’ is higher than ‘managers’ and ‘Employees’ to restrict them from reading prediction data. |
| **4.** | **Managerial** | | Edit to Employee data | Bell-Lepadula, Biba Model. | Assumption: Admin has granted few accesses to Managers as follows:   * Managers can view and edit their personal data. * Managers can view their salary but can’t edit it. * Managers can view/edit employee personal data. * Managers can view/edit employees' salaries.   Managers can edit their data and employees data, this can be achieved using the Biba model by assigned Managers higher classification than Employees. Similarly Bell-Lapadula model is used to allow managers to read their data and employees data also under the same classification. |
|  |  | | Receive privileges from administrator | Biba and Bell-Lepadula. | We assumed the privileges of Managers as follows:   * Managers can view and edit their personal data. * Managers can view their salary but can’t edit it. * Managers can view/edit employee personal data. * Managers can view/edit employees' salaries.   Based on these assumptions, we require a hierarchy of roles which can be easily achieved by using a combination of Biba and Bell-Lapadula models. We defined the classification and set of categories for Bell-Lepadula , and Biba model. |
| **5.** | **Role Hierarchy** | |  | Bell-Lapadula Model. | We have ‘Admin’, ‘Customer’, ‘Manager’, ‘Employee’ roles in our case. To classify these into different roles so that confidentiality and integrity is maintained, Bell-Lapadula model is highly useful. Bell\_Lapadula defines ‘Classification’ levels and user defined ‘Set of Categories’ which can be used to achieve our goals in this case. |

Subjects: Admin, Customer, Manager, Employee.

Objects: adminData, predictionData, managerData, employeeData, empSal, manSal.

I have classified subjects vans objects according to Bell-Lapadula model as follows:

classification = {"Admin":3, "Customer":2, "Manager":1, "Employee":0, "adminData":3, "predictionData":2, "managerData":1, "employeeData":0, "empSal":0, "manSal":1}

I have assigned the user defined categories as follows:

Here note that the customer is assigned a different category so that he can’t read/edit other user details. As classification of ‘Customer’ is higher than ‘Employee’ and ‘Manager’, ‘Customer’ is assigned a lower category so that it does not view/edit the employee or manager details.

category = {"Admin":1, "Manager":1, "Employee":1, "adminData":1, "predictionData":0, "managerData":1, "employeeData":1, "empSal":1, "manSal":1, "Customer": 0}

Above classification and categories are for ensuring the confidentiality property. It ensures proper subjects are allowed to read the objects which they have access to.

Following function f represents the clearance level and current security level of different subjects.

f = {"Admin":[3, 1], "Customer":[2, 2], "Manager":[1, 0], "Employee":[0,0]}

Now to ensure that ‘edit’ operation is done properly, we have implemented the Biba model. Following is the modified classification and set of categories to ensure integrity.

classification\_i = {"Admin":3, "Customer":2, "Manager":1, "Employee":0, "adminData":3, "predictionData":2, "managerData":1, "employeeData":0, "empSal":1, "manSal":3}

Note classification of employee salary and manager salary changes so that only admin can edit manager salary and manager can edit employee salary.

category\_i = {"Admin":1, "Manager":1, "Employee":1, "adminData":1, "predictionData":2, "managerData":1, "employeeData":1, "empSal":1, "manSal":1, "Customer": 0}

Note that the category of ‘predictionData’ changes here. This is because no subject has edit access to prediction data.

As it is mentioned in the question that admin has only read access to ‘Prediction Data’, we have assumed that no other subjects have write access to prediction data.

Following is the access control matrix as per given scenario:

Access Control Matrix:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Admin Data | Manager Data | Employee Data | Manager Salary | Employee Salary | Prediction Data |
| Admin | read, write | read, write | read, write | read, write | read, write | read |
| Customer | - | - | - | - | - | read |
| Manager | - | read, write | read, write | read | read, write | - |
| Employee | - | - | read, write | - | read | - |