**Security Policy**

So, there’s good news and bad news regarding security policy. The good news is that a very simple security policy will facilitate OS-level protection of all this data within the bounds of the program. The bad news is that I couldn’t, from a cursory glance, determine how easy it is to implement a custom security policy with SELinux. This will require some research and tinkering.

Rules:

* A user cannot read any data whose category range is not a subset of their category range.
  + User with category range c1.c2 cannot read data with range c2.c3
* A user can read any data whose category range is a subset of their range
  + User with category range c1.c2 can read data with range c1
* A user can only write to data inside their category range and at/below their security level
  + User with context s1:c1.c2 can write to data with context s0:c1

Diagram for user with context s1:c1.c2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | c0 | c1 | c2 | c3 |
| s2 | Access denied | Read-only | | Access denied |
| s1 | R/W | |
| s0 |

**Security constraints of the mock system:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Students | Instructors | Coordinators |
| Can read | All student account data [UC2] Instructor name [UC6] All course-student mapping data [UC6]  All course data [UC6] | All instructor account data [UC2]  Student name [UC6]  Course-student mapping data [UC5]  All course data [UC6] | Everything |
| Can write | Nothing | Course-student mapping grade column [UC5]. | Everything [UC3, UC4] |

* Students
  + Must be able to view, but not edit their account details, which are stored in the students table.
  + Must be able to view their schedule (including grades). This data is stored in the course\_student\_mappings and courses tables.
    - They should be able to see instructor names since it’s useful to see what instructor is teaching the class they’re taking
* Instructors
  + Must be able to view their account details, which are stored in the instructors table
  + Must be able to view their schedule, stored in the courses table.
  + Must be able to view and edit student grades, stored in the course\_student\_mappings table
  + Student names should be visible to instructors for seeing the students in their roster
* Coordinators
  + Even though coordinators aren’t listed as being able to modify everything in the requirements, they don’t need to be denied access to any data (this will make it easier for us to implement.

Using the policy defined above with these access restrictions, we can create the following table of security contexts:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | c0 (visible to instructors) | c1 (visible to instructors and students) | c2 (visible to students) | c3 (visible to coordinators only) |
| s2 (writeable by coordinators only) | instructors | courses, course\_student\_mappings, students.name, instructors.name | students | coordinators |
| s1 (writeable by coordinators and instructors) |  | course\_student\_mappings.grade |  |  |
| s0 (writeable by students) |  |  |  |  |

Text in black represents a database table. Text in blue represents a database column.

Note that coordinators have R/W access to data in every cell of this table.

From this table, it can be derived that:

Coordinators have security context s2:c0.c3

Instructors have security context s1:c0.c1

Students have security context s0:c1.c2