Questionnaire

See Lecture 10 on Role-Based Access Control

1. Which modelling languages are specifically extended to model role-based access control?
   - Secure Tropos
   - Secure intentional distribution
   - KAOS extension to security
   - Misuse cases
   - Mal-activity diagrams
   ☒ UMLsec

2. How is the process of limiting access to the resources of a system only to authorised programs, processes or other systems called?
   - Access
   ☒ Access control
   - Permission assignment
   - Session

3. What are the main security actions?
   ☒ Insert / create
   ☒ Select / read
   ☒ Change / update
   ☒ Remove / delete

4. How is a relationship among roles called?
   - Sessions
   - Permission hierarchy
   - Role hierarchy
   ☒ Constraint

5. In order to strengthen security permissions, authorisation constraints are defined in
   - Security action models;
   ☒ SecureUML models;
   - UMLsec models;
   - SecureUML and UMLsec models
   - Transformation models

6. In SecureUML protected Operations are defined using
   - Class stereotypes
7. What are the major tasks of the system administrator?

☒ Manage users and roles
☒ Create assignment relationships
☐ Predefine secured operations and objects
☒ Establish relationships between roles, secured operations and objects

8. Which stereotype indicates that UMLsec model presents the role-based access control policy?

☐ <<secuml.role>>
☒ <<rbac>>
☐ {protected}
☐ {right}

9. What models are used to define access control policies?

☒ Attribute-based access control
☐ Unicorn-oriented privileges
☒ Risk-adaptive access control
☐ Bitcoin-oriented access assignment

10. How is a person who interacts directly with the computer system called?

☐ Administrator
☒ User
☐ Subject
☐ Role
Exercise 1

The SecureUML model (see Fig. 1) was received by transforming the corresponding UMLsec model. Complete this SecureUML model by introducing missing language constructs. Do not forget (!) to define and apply needed security actions.

10 points
Solution is provided in Demo D002
Exercise 2

The UMLsec model (see Fig. 2) was received by transforming the corresponding SecureUML model. Complete this UMLsec model by introducing missing language constructs.

10 points

<table>
<thead>
<tr>
<th>MeetingAgreement</th>
<th>FootballFederationEmployee</th>
<th>TeamRepresentative</th>
</tr>
</thead>
<tbody>
<tr>
<td>setTimePlace</td>
<td>enterAgreement Details</td>
<td>getAgreement Information</td>
</tr>
<tr>
<td>changeTime Place</td>
<td>changeMeeting Info</td>
<td></td>
</tr>
<tr>
<td>viewTime Place</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2. UMLsec model
Solution is provided in Demo D001

```ruby
{protected = (setTimePlace)}
{role = (Bob, FootballFederationEmployee)}
{right = (FootballFederationEmployee, setTimePlace)}

{protected = (changeTimePlace)}
{role = (Bob, FootballFederationEmployee)}
{right = (FootballFederationEmployee, changeTimePlace)}

{protected = (viewTimePlace)}
{role = (Marie, TeamRepresentative)}
{right = (TeamRepresentative, viewTimePlace)}

{protected = (viewTimePlace)}
{role = (Karl, TeamRepresentative)}
{right = (TeamRepresentative, viewTimePlace)}
```