EXAM TASKS

[The sum of points equals to 100]

PART I: Security Risk Management

Problem description: Mobile banking is a term used for performing balance checks, account transactions, payments, credit applications etc. via a mobile device such as a mobile phone or Personal Digital Assistant (PDA). The earliest mobile banking services were offered via SMS. With the introduction of the first primitive smart phones with WAP support enabling the use of the mobile web in 1999, the first European banks started to offer mobile banking on this platform to their customers. Mobile banking can offer account information services such as:

- Mini-statements and checking of account history;
- Alerts on account activity or passing of set thresholds;
- Monitoring of term deposits;
- Access to loan statements;
- Access to card statements;
- Mutual funds / equity statements;
- Insurance policy management;
- Balance checking in the account;
- Recent transactions;
- Due date of payment (functionality for stop, change and deleting of payments);
- PIN provision, Change of PIN and reminder over the Internet;
- Blocking of (lost, stolen) cards.

A specific sequence of SMS messages would enable the system to verify if the client has sufficient funds in his or her wallet and authorize a deposit or withdrawal transaction at the agent.

To secure mobile banking, some developers have applied the security patterns. One of the security risk-oriented patterns is provided in Table 1. This pattern partially describes how to secure data transmission between two system entities.

Use the security risk-oriented language (choose one among security risk-oriented BPMN, Secure Tropos, Misuse cases diagrams, or Mal-activity diagrams) and illustrate (one example is enough!) how this pattern is applied in the given context of the mobile banking.

Three diagrams needs to be created in order to show:

- **Task 1**: What are the context, assets and their security criteria; (15 points)
- **Task 2**: How does the security risk constitute itself; (20 points)
- **Task 3**: What is the countermeasure (i.e., risk treatment, security requirements, and controls), and how does it mitigate the security risk. (15 points)

Note: Although a picture is worth 1000 words, short textual explanations of the diagrams are welcome!
**Table 1**: Security risk-oriented pattern – securing data between two entities

<table>
<thead>
<tr>
<th>Assets and security objective definition</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Business assets</td>
<td>Data submitted, transmitted and employed</td>
</tr>
<tr>
<td>IS assets</td>
<td>Input interface, Transmission medium that transfers data, and Server</td>
</tr>
<tr>
<td>Security criteria</td>
<td>Confidentiality of data</td>
</tr>
</tbody>
</table>

**Risk definition**

<table>
<thead>
<tr>
<th>Risk name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk</td>
<td>An attacker intercepts the transmission medium due to its characteristics of being intercepted and misuses the data leading to loss of data confidentiality</td>
</tr>
</tbody>
</table>
| Impact    | - Harm data submitted and employed in server  
- Loss of reliability of the transmission medium  
- Negation of data confidentiality |
| Event     | An attacker intercepts the transmission medium due to its characteristics to be intercepted and misuses the data. |
| Threat    | An attacker intercepts the transmission medium and misuses the data. |
| Vulnerability | - Characteristics of transmission medium to be intercepted;  
- Lack of crypto-functionality at input interface and server. |
| Threat agent | An attacker with means to intercept transmission medium by acting as a proxy. |
| Attack method | - Intercept transmission medium by establishing a proxy between input interface and server;  
- Misuse data by capturing, reading and keeping data for the later use. |

**Risk treatment definition**

<table>
<thead>
<tr>
<th>Risk treatment</th>
<th>Security requirement</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk reduction</td>
<td>Make data unreadable to attackers</td>
<td>Cryptographic algorithm</td>
</tr>
<tr>
<td>Risk avoidance</td>
<td>Change the transmission medium so that there will be no possibility to intercept it using proxy.</td>
<td>Physically delivery of data;</td>
</tr>
</tbody>
</table>

**PART II: Model-driven Security**

**Problem description**: Alice is a security analyst at the SDO company, which is a software development organisation specialised in creating new information management solutions. In order to understand requirements for the Course Admission System (CAS), firstly, Alice has interviewed four people: Eva, John, Jeremy, and Tom. Next, she has created a security model using UMLsec modelling language (see Figure 1; Tables 2 presents the association tags; and Table 3 gives short description of few actions). However, now she needs (i) to validate the elicited security requirements with Eva, John, Jeremy, and Tom; and (ii) to prepare another model, such that SDO implementation team could generate the executable CAS.

**Task 4**: Using Figure 1 or/and model created in Task 5, help Alice to specify at least 5 CAS security requirements textually. Each requirement should say what the CAS needs to do, it should be unambiguous, cohesive and testable.  

(10 points)

Requirement **ID1**: The CAS application …

Requirement **ID2**: …
**Task 5:** Using the SecureUML modelling language, create the class diagram, which would present the RBAC policy in correspondence to the Alice model.

The class diagram should specify:
- Protected Resource(s), its Attributes and protected Operations (10 points)
- Users, Roles, and User assignments (5 points)
- Permissions, Permissions assignment and Security actions (15 points)
- Security authorisation constraints, either in OCL or at least in plain text (10 points)

Figure 1: Application Management Process
## Table 2: Associated tags

<table>
<thead>
<tr>
<th>Protected</th>
<th>Role</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create new application</td>
<td>John, Student</td>
<td>Student, Create new application</td>
</tr>
<tr>
<td>Display accepted application</td>
<td>Tom, ScholarshipCommittee</td>
<td>ScholarshipCommittee, Display accepted application</td>
</tr>
<tr>
<td>Display submitted application</td>
<td>Eva, AdmissionCommittee</td>
<td>AdmissionCommittee, Display submitted application</td>
</tr>
<tr>
<td>Add scholarship decision</td>
<td>Tom, ScholarshipCommittee</td>
<td>ScholarshipCommittee, Add scholarship decision</td>
</tr>
<tr>
<td>Add decision about application</td>
<td>Eva, AdmissionCommittee</td>
<td>AdmissionCommittee, Add decision about application</td>
</tr>
<tr>
<td>Display student’s application</td>
<td>John, Student</td>
<td>Student, Display student’s application</td>
</tr>
</tbody>
</table>

## Table 3: Action description

<table>
<thead>
<tr>
<th>Action …</th>
<th>… concerns attributes…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create new application</td>
<td>(created and filled by the student) Contact information, Previous studies, Requested course; (left with the null value, but created as part of the application) Acceptance decision, Scholarship decision</td>
</tr>
<tr>
<td>Display submitted application</td>
<td>Previous studies, Requested course</td>
</tr>
<tr>
<td>Add decision about application</td>
<td>Acceptance decision</td>
</tr>
<tr>
<td>Display accepted application</td>
<td>Contact information, Previous studies, Requested course, Acceptance decision</td>
</tr>
<tr>
<td>Add scholarship decision</td>
<td>Scholarship decision</td>
</tr>
<tr>
<td>Display student’s application</td>
<td>Contact information, Requested course, Acceptance decision, Scholarship decision</td>
</tr>
</tbody>
</table>
**SOLUTIONS**
(alternative solutions are possible)

**PART I: Security Risk Management**

Instantiation of the pattern:

<table>
<thead>
<tr>
<th>Assets and security objective definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business assets</strong></td>
</tr>
<tr>
<td><strong>IS assets</strong></td>
</tr>
<tr>
<td><strong>Security criteria</strong></td>
</tr>
</tbody>
</table>

**Risk definition**

| **Risk** | An attacker intercepts the mobile wireless network due to its characteristics of being intercepted and misuses the query for card statement leading to loss of confidentiality of the query for card statement. |
| **Impact** | - Harm query for card statement submitted and employed in the bank server; - Loss of reliability of the mobile wireless network; - Negation of confidentiality of the query for card statement. |
| **Event** | An attacker intercepts the mobile wireless network due to its characteristics to be intercepted and misuses the query for card statement. |
| **Threat** | An attacker intercepts the transmission medium and misuses the data. |
| **Vulnerability** | - Characteristics of mobile wireless network to be intercepted; - Lack of crypto-functionality at WAP browser and bank server. |
| **Threat agent** | An attacker with means to intercept the mobile wireless network. |
| **Attack method** | - Intercept mobile wireless network by establishing a proxy between WAP browser and bank server; - Misuse query for card statement by capturing, reading and keeping it for the later use. |

**Risk treatment definition**

| **Risk treatment** | Risk reduction | Risk avoidance |
| **Security requirement** | - Make of the query for card statement unreadable to attackers | Change the mobile wireless network so that there will be no possibility to intercept it. |
| **Control** | Cryptographic algorithm | Physically delivery of the query for card statement. |

**Task 1:** What are the context, assets and their security criteria;
Task 2: How does the security risk constitute itself;

Task 3: What is the countermeasure (i.e., risk treatment, security requirements, and controls), and how does it mitigate the security risk.

Risk reduction:
Risk avoidance:

PART II: Model-driven Security

Task 4: Using Figure 1 or/and model created in Task 5, help Alice to specify at least 5 CAS security requirements textually. Each requirement should say what the CAS needs to do, it should be unambiguous, cohesive and testable.

All five requirements are of the authorisation requirements type. They describe what access different roles have to the application.

- Requirement ID1: The CAS should allow the student to create new application. Student should be able to fill in the contact information, previous studies and requested course.

- Requirement ID2: The CAS shall display the newly created student’s application (i.e., the previous studies and requested course) to the Admission Committee.

- Requirement ID3: The CAS shall allow Admission Committee to enter decision about admission to the requested course.

- Requirement ID4: The CAS shall display the accepted application (i.e., contact information, previous studies, requested course, acceptance decision) to the Scholarship Committee.

- Requirement ID5: The CAS shall allow Scholarship Committee to add decision about the scholarship assignment.

Task 5: Using the SecureUML modelling language, create the class diagram, which would present the RBAC policy in correspondence to the Alice model.

The class diagram should specify:

- Protected Resource(s), its Attributes and protected Operations (10 points)
- Users, Roles, and User assignments (5 points)
- Permissions, Permissions assignment and Security actions (15 points)
Creativity – your key to secure software!!!

First two authorisation constraints are placed on the permissions of Jeremy:

- **AC1**: A permission should be granted to Jeremy as an *AdmissionCommittee* member to display submitted application (operation *DisplaySubmittedApplication*), but he should be restricted from adding decision about application (operation *AddDecisionAboutApplication*).
- **AC2**: A permission should be granted to Jeremy as a *ScholarshipCommittee* member to add scholarship decision (operation *AddScholarshipDecision*), but he should be restricted from viewing application (operation *DisplayAcceptedApplication*).

Other constraints are about relating the permitted actions, protected operations and attributes of the protected resource:

- **AC3**: Attribute *fillInApplication*:CREATE of the *StudentPermission* is associated with *Application* operation *CreateNewApplication()* over attributes *ContactInformation*, *PreviousStudies*, *RequestedCourse*, *AcceptanceDecision*, and *ScholarshipDecision*.
- **AC4**: Attribute *viewApplication*:SELECT of the *StudentPermission* is associated with *Application* operation *DisplayStudentApplication()* over attributes *ContactInformation*, *RequestedCourse*, *RequestedCourse*, *AcceptanceDecision*, and *ScholarshipDecision*.
- **AC5**: Attribute *viewApplication*:SELECT of the *ApplicationCommitteePermission* is associated with *Application* operation *DisplaySubmittedApplication()* over attributes *PreviousStudies*, and *RequestedCourse*.
- **AC6**: Attribute *makeAcceptanceDecision*:UPDATE of the *ApplicationCommitteePermission* is associated with *Application* operation *AddDecisionAboutApplication()* over attribute *AcceptanceDecision*.
• **AC7**: Attribute `viewApplication:SELECT` of the `ScholarshipCommitteePermission` is associated with `Application` operation `DisplayAcceptedApplication()` over attributes `ContactInformation`, `PreviousStudies`, `RequestedCourse` and `AcceptanceDecision`.

• **AC8**: Attribute `decideWhtherToAssignScholarship:UPDATE` of the `ScholarshipCommitteePermission` is associated with `Application` operation `AddScholarshipDecision()` over attribute `ScholarshipDecision`. 
BONUS – Multiple Choice Questionnaire

[Each correctly answered question gives you 1 point.]

Q1. How is the potentiality of risk event estimated?
   a) Estimated as maximum impact level of concerned impacts;
   b) Estimated as security needs (the same method, just reversed thinking);
   c) Estimated from the security needs;
   d) Estimated as a sum of threat likelihood and vulnerability level;
   e) All a, b, c, and d;
   f) **Neither a, b, c, nor d.**

Q2. Which security problems could be caused by the API abuse?
   a) Buffer overflows;
   b) Privacy violation;
   c) Setting manipulation;
   d) **Directory restriction:**
   e) All a, b, c, and d;
   f) Neither a, b, c, nor d.

Q3. What are identification requirements?
   a) When application protects itself from infection by unauthorised programs;
   b) When business keeps its sensitive data private from unauthorised use;
   c) When application ensures that its data are not intentionally corrupted;
   d) **When business shall identify its externals before interacting:**
   e) All a, b, c, and d;
   f) Neither a, b, c, nor d.

Q4. What are misuse cases?
   a) **It is a security modelling technique:**
   b) They describe what the system should avoid from doing;
   c) They describe what the system should do to meet the objectives;
   d) They describe what the systems should do related to the threats;
   e) All a, b, c, and d;
   f) Neither a, b, c, nor d.

Q5. What is an RBAC role?
   a) Any person who interacts with a system;
   b) A relationship among sessions;
   c) An entity that causes information to flow;
   d) A specific type of interaction between a subject and an object;
   e) All a, b, c, and d;
   f) **Neither a, b, c, nor d.**
Q6. Which RBAC functionality should be supported, after user assignments and permission assignments are defined?

a) Transformation of SecureUML diagram to UMLsec;
b) **View of assigned users, roles and their permissions**;
c) **View session roles and permissions**;
d) Creation of sessions and checking the access;
e) All a, b, c, and d;
f) Neither a, b, c, nor d.

Q7. What is shoulder surfing?

a) Pawing through a target’s garbage in search of valuable information;
b) **Stealth observation of the target to obtain or deduce confidential information**;
c) Exploiting user’s curiosity to deliver malware;
d) Continuing to function even if a data centre is destroyed;
e) All a, b, c, and d;
f) Neither a, b, c, nor d.

Q8. Which of these “points” do not belong to the “Seven Security Touchpoints”?

a) **Maturity of test plans**;
b) Security requirements;
c) Risk-based security tests;
d) Penetration testing;
e) All a, b, c, and d do not belong;
f) All a, b, c, and d do belong.

Q9. Which attack methods could be used by threat agents to break into the system?

a) SQL injection attack;
b) Pre-texting or phishing;
c) Seamless installation of the key-loggers;
d) Cross-site scripting attack;
e) **All a, b, c, and d**;
f) Neither a, b, c, nor d.

Q10. What is a security criterion?

a) Anything that has value to the organisation;
b) **A property or constraint on business assets that characterises their security needs**;
c) A characteristic of an IS asset that can constitute a weakness or a flaw in terms of IS security;
d) Potential attack, carried out by an agent;
e) All a, b, c, and d;
f) Neither a, b, c, nor d.