Security Model for Health Care Computing and Communication Systems

Anas ABOU EL KALAM, Yves DESWARTE
(Anas, deswarte)@laas.fr

Plan

- System security requirement definition
- Concepts for HCCS security policies
- Security model
- Conclusions and perspectives

Plan

- System security requirement definition
  - Sensitive information
  - Risks
  - Security requirements
- Concepts for HCCS security policies
- Security model
- Conclusions and perspectives
### Security Model

**HCCS Security** (Security, Safety, Quality, Privacy)

- **Confidentiality**
  - The non-occurrence of unauthorized disclosure

- **Integrity**
  - The non-occurrence of inadequate alterations

- **Availability**
  - The readiness for usage

- **Accountability**
  - (Availability & Integrity) [operation, identity of the person who realized the operation ...]
  - (Availability & Integrity & confidentiality) [information + meta-information]

### Conclusions and Perspectives

- **System security requirement definition**
- **Concepts for HCCS security policies**
  - Groups of objects
  - Roles in Teams
  - Context

- **Security model**

- **Conclusions and perspectives**
Roles / Groups of objects

Role: structure subjects
Role r: Permission p
User

Group of Objects (GO): structure objects
Action a GO go
Object m

Role: associate subjects that fulfill the same functions
GO: set of objects that satisfy a common property
- Logical criteria based on access rights
- Objects on which the same actions are realized
- E.g., patient of the unit X

Example

Resource-Unit-C5: Resource-Surgical-Unit: Clinical-resource
Surgical-Ward: Ward 2
Post: computer 10
Speciality-Record: PMF n

Groups of objects: advantages

- Facilitate security policy management/expression/comprehension
- Update easily the security policy when new subjects/objects are added to the system

Reduce complexity of access rights

Action 1
Object 1

Action NA
Object NO

Cost: N_A*N_O

Reduce administration errors

<table>
<thead>
<tr>
<th>relation</th>
<th>Aspect</th>
<th>Managed by</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Role, Permission), (action, group)</td>
<td>Relatively stable</td>
<td>Administrator</td>
</tr>
<tr>
<td>{User, role}, {objects, group}</td>
<td>Altered more often</td>
<td>Reception staff ...</td>
</tr>
</tbody>
</table>

Roles in Teams

Some differences with TMAC
- TMAC: binary relations: (user, role), (role, team)
- A user can activate any one of his roles in any one of his teams
- Our approach: ternary relation: (user, role, team)
- A user can activate a subset of his roles in each of the teams which he participates in

Why?
- For the same role, authorizations can differ from an organization to another
- The same user can activate a subset of his roles in each of the teams which he participates in

Role in Team

<table>
<thead>
<tr>
<th>Role</th>
<th>Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>RoleTeam</td>
</tr>
<tr>
<td>Bob: Physician</td>
<td>PhysicianCS: RoleTeam</td>
</tr>
<tr>
<td>NurseCS: RoleTeam</td>
<td></td>
</tr>
</tbody>
</table>

Context

Context of a role:
- Cardinality, static/dynamic mutual exclusion

Context of objects:
- Duration attribute for the storage of certain data; location attribute

User attributes:
- Specific authorization, temporary rights

Context of use:
- Healthcare process
- Purpose of use

Every access must belong to one of two cases
- the user’s team participates in the process treating the patient
- particular situation: declaration of a purpose

the activity of care has been already defined by authorized persons
### Alphabet of the language

- **Constants**
  - Instances of security policy entities: users, roles, teams ...

- **Variables**
  - E.g.: \( u \in \text{Users} \) (\( u \) is a variable of type User), \( r \in \text{Roles} \), \( g \in \text{Groups of Objects} \)

- **Functions**
  - Describing/building terms \( \Rightarrow \) deriving information about their properties
  - \( \text{PMF} \): patient identity, \( \text{CP} \): identity, nurse report, diagnosis, prescriptions \( \Rightarrow \) Record

- **Predicate**
  - E.g.: \( \text{AUR}(u, r, t) \) is a relation symbol of type (user, role, team)

- **Actions**
  - E.g.: \( \text{TRANSMIT}(u, f, u') \), \( \text{CREATE}(u, t) \)

### The language

\[
\begin{align*}
\mathcal{A}(t_1, \ldots, t_n) & : = \text{Predicate}(t_1, \ldots, t_n) \mid \text{Action}(t_1, \ldots, t_n) \\
\mathcal{F} & : = \mathcal{A}(t_1, \ldots, t_n) \mid \neg \mathcal{F} \mid \mathcal{F} \bigwedge \mathcal{F} \mid \mathcal{F} \bigvee \mathcal{F} \mid \mathcal{O}(\mathcal{F}) \mid \mathcal{P}(\mathcal{F}) \mid \mathcal{F} \bigcup \mathcal{F} \mid \mathcal{F} \bigcap \mathcal{F}
\end{align*}
\]

---

### Truth conditions

- \( M, w \models \mathcal{O}(f) \Rightarrow [\forall w, r \mathcal{R}(w)] \Rightarrow M, w \models \mathcal{F} \)
  - \( f \) is true in every world \( w \) which \( w \) is in relation with (... in all possible worlds, ...)

- \( M, w \vdash \mathcal{P}(f) \Rightarrow [\exists w, r \mathcal{R}(w)] \Rightarrow M, w \models \mathcal{F} \)
  - It should be possible to reach a world in which \( f \) is true (... in some possible cases)

- \( M, w \models \mathcal{F} \Rightarrow [\forall w, r \mathcal{R}(w)] \Rightarrow M, w \not\models \mathcal{F} \)
  - None of the accessible worlds should allow to conclude that \( f \) is true (... none of the ...)

### System description

**What?**

The functional aspects of the system that are relevant for security

**How?**

- Propositional logic operators
- Define the internal structure of the worlds
  - E.g.: \( q \Rightarrow r \) means in any world \( w \) where \( q \) is true, \( r \) is also true

**Example**

Role hierarchy

- \( \text{AR}(u, \text{physician}) \Rightarrow \text{AR}(u, \text{Clinical Staff}) \)
- \( \text{AR}(u, \text{nurse}) \Rightarrow \text{AR}(u, \text{Clinical Staff}) \)
- ...

---

### Security properties

**What?**

Express security requirements (confidentiality, integrity and availability)

**How?**

- Modal operators \( (P/O/R/F) \)
- Allow modifying the properties of the accessibility relations between the worlds

**Example**

\( \neg \text{AR}(u, \text{pharmacist}) \wedge \text{CREATE}(u, \text{prescription}) \)

- \( \Rightarrow \) forbids a pharmacist to create prescriptions
- \( \Rightarrow \) none of the possible evolutions of the system should allow to conclude that a pharmacist can create a prescription

---
Security rules

What?
Regulations ...

How?
- Modal formulas with at least one non modal clause
- Describe the link between the P/O/R/F and the state of the system

Example
- \( \text{ARec}(p, \text{Record}) \Rightarrow \text{P}[\text{READ}(p, \text{Record})] \)
  - \( \text{ARec}(p, \text{Record}) \): a predicate that associates each patient to his/her medical record
  - \( \Rightarrow \): every patient is permitted to read his/her medical record

\[ \text{ARec}(\text{Betty}, \text{Record}) \rightarrow \text{W1} \]
\[ \text{W} \rightarrow \text{W2} \rightarrow \text{READ(Betty, Record)} \]

Conclusions and perspectives

- **What to protect?** against whom/what? what do we need?
  - **Security policy**
    - Structure objects according to access right criterias \( \Rightarrow \) *group of objects*
    - Job performed by the user \( \Rightarrow \) *Role*
    - The relation between the HCP and the patient \( \Rightarrow \) *team/organization*
    - Normal access: strong authorization \( \Rightarrow \) *process of care*
    - Flexibility and accountability \( \Rightarrow \) *Purpose of use*

- **Security model**
  - Extending/adapting deontic logic \( \Rightarrow \) \( P/O/R/F + \) actions
  - Describing the system, expressing the security properties and the security rules

- **Reasoning** about our security model by using analytic tableaux
  - **Implementation** by using security mechanisms such as distributed capabilities (MAFTIA) or XML interpretation (univ. of Milan) ...

Questions

Wednesday, 28 May 03 IFIP SEC 03, Athens