Or-BAC: Organization based Access Control

Presented by:
Anas Abou El Kalam (LAAS-CNRS)

Advisor:
Yves Deswarte

Joint work by:
P. Balbiani, S. Benferhat, F. Cuppens, R. Elbaida, A. Miege, C. Saurel, G. Trouessin

Motivation

Security Policies?

- Rules that satisfy contextual permissions/prohibitions
  - e.g., the context of urgency
- Rules that satisfy P/F/O/R
- Rules that are specific to the organization
  - Each sub-organization may have its own SP, ...

Plan

- Examples of basic models/Policies
- Or-BAC: the model
- A language to specify an Or-BAC SP
- An example: a case study in HCCCS
- Conclusions and perspectives

Plan

- Examples of basic models/Policies
  - DAC & HRU
  - RBAC
- Or-BAC: the model
- A language to specify an Or-BAC SP
- An example: a case study in HCCCS
- Conclusions and perspectives
Basic models: HRU

**Subjects:** active entities

**Objects:** (active + non-active) entities

**Actions**

![Diagram of HRU model]

But... 
- When new subjects/objects/actions are introduced in the system?
  - the updating of the SP is quite complicated!!
- Only permissions
  - F/O/R are not included!!

Basic models: RBAC

**Role:** structure subjects

- The concept of permission is primitive
  - For a given application, the RBAC model must be refined to make explicit the structure of permissions
- For the same role, authorizations can differ from an organization to another
- How to specify a P that depends on a given context?
  - All users that play a certain role will inherit the role permissions ...
  - but, if the physician does not treat the patient?

Plan

**Examples of basic models/Policies**

**Or-BAC:** the model

- Subjects & Roles
- Objects & Views
- Actions & Activities
- The context

A language to specify an Or-BAC SP

An example: a case study in HCCCS

Conclusions and perspectives
Or-BAC: Objects and views

- **Role**: structure subjects
  - User i
  - Role r
  - Permission p
  - Role: associate subjects that fulfill the same functions
    - Makes easier the update of the SP when new subjects are added to the system

- **View**: structure objects
  - View v
  - Object n
  - View: set of objects that satisfy a common property
    - Logical criteria based on access rights
    - Characterize the ways objects are used in Orgs
    - Makes easier the update of the SP when new objects are added to the system

Or-BAC: Actions and activities

- **Action**: File Operations such as read, write, send, ...

- **Activity**: To abstract actions
  - To join actions that share the same principles
  - To be able to characterize organizations that structure differently the same activities

Or-BAC: Contexts

- **What?**
  - Specify the concrete circumstances where organizations grant role permissions to perform activities on views
  - E.g., urgency, attending physician, etc.

- **Example**
  - Define(Hospital A, Bob, F1.doc, read, Urgency) = within the organization "hospital A", the context "urgency" is true between the subject "Bob", the object "F1.doc" and the action "read".

The conditions required for a given context are formally specified by logical rules

Or-BAC: the model

Concrete Authorizations

- Subject
- Object
- Action
- Elementary object
- User
- Organization
- Activity
- View
- Context
- SP level

Abstract entities
Plan

- Examples of basic models
- Or-BAC: the model
- Language to specify an Or-BAC SP

Why & What
Alphabet
The language
Axioms

An example: a case study in HCCCCS

Conclusions and perspectives

---

Formal System: the language

What?
- A first-order language

Why?
- Representation and reasoning about P/O/R/F

Alphabet of the language
- Constants
  - Instances of security policy entities: users, roles, organizations ...
- Variables
  - E.g.: u ∈ Users (u is a variable of type User), r ∈ Roles ...
- Predicate
  - E.g.: EMPLOY (org, s, r) is a relation symbol of type (Organization, Subject, Role)
- Functions
  - Describing/building terms => deriving information about their properties
    - E.g., Attending-physician(p)

The language
- $f ::= \neg f \mid f \land f \mid f \lor f \mid \forall x f \mid \exists x f$

---

Formal System: axioms

- Employ(org, s, r) ∧
  - If org employs subject s in role r,
  - if org uses object a in view v,
  - if org considers that a falls within the activity a
  - if within org, the context c is true between s, o, a
  - if within c, org grants r permission to perform a on v
  - then s has permission to perform a on o.

- Use(org, o, v) ∧
  - Consider(org, a, o) ∧
  - Define(org, s, a, c) ∧
  - Permission(org, r, v, o, c)
  - ⇒ Is_Permitted(s, o, a)

- Obligation → Recommendation
  - Every obligation is also a recommendation

- Recommendation → Permission
  - Every recommendation is also a permission

---

Plan

- Examples of basic models
- Or-BAC: the model
- Language to specify an Or-BAC SP
- An example: a case study in HCCCCS

Examples of rules
- System description
- Security rules

Conclusions and perspectives
System description: Subjects & Roles

- John (Director)
- Mary (Admin_Assistant)
- Paul (Surgeon)
- Peter (Nurse)
- Max (Anesthetist)

Employ(Hosp1, John, director)
Employ(Hosp1, ST1, Surgical_team)
Employ(ST1, Peter, Nurse)
Employ(RT2, Peter, Radiologist_assistant)

System description: Objects & Views

- Use(Org2, f31.doc, Medical_record)
- Use(Hosp1, f1.txt, Medical_record)

System description: Actions & Activities

- Consulting
  - Select
  - OpenWord
- Writing
  - Update

System description: Context

∀x ∀y ∀α
Define(ST1, s, o, α, Attending_team) ↔
∃r Employ(ST1, s, r) ∧ Name(o) ∈ Patient(s)

In ST1, the context Attending_team is true between s, o, and α if s play some role in ST1 and o is a record corresponding to one of the patients treated by ST1.
Security policy

- Permission(RT2, Physician, Medical_record, Consulting, attending_team) 
  $\Rightarrow$ RT2 permits physicians to consult a medical record if this record 
  correspond to a patient of RT2 

  $\Rightarrow$ The permissions associated with the physician role can change from one 
  organization to another and the respective context may be also different

Hierarchies

$\forall$s $\exists r_1, r_2$ 

Permission(ST1, r1, v, a, c) $\Rightarrow$ Permission(ST1, r2, v, a, c) 

$\Rightarrow$ inheritance of permissions, within ST1 between a role r1 (e.g., physician) and 

role r2 (e.g., surgeon) 

$\Rightarrow$ We can specify that inheritance between two given roles only applies to a given 

org and would be false in another one

Constraints

$\forall$s 

$\neg$Employ(Org1, s, surgeon) $\land$ Employ(Org1, s, Anesthetist) 

$\Rightarrow$ in org1, no subject s can be employed both as a surgeon and an anesthetist

Thursday 5th June 2003  
Policy 03, Lake Como  
21

Conclusions and perspectives

Or-BAC

- Central concept (SP are parameterized by the org) $\Rightarrow$ organization
- How this org is employing subjects $\Rightarrow$ Role
- How this org is using objects $\Rightarrow$ View
- How this org is performing actions $\Rightarrow$ Activity
- How this org is defining contexts $\Rightarrow$ The relationship "Define"

Security Policy

- P/O/F/R (organization, role, view, activity, context) 
- A first-order language

Present work

- Resolving conflicting situations (I.e., P & O \(\parallel\) P & F) $\Rightarrow$ possibilistic logic
- Administrating the SP defined in our model $\Rightarrow$ ARBAC02
- Implementing our SP $\Rightarrow$ "XML" or "distributed capabilities"

Thursday 5th June 2003  
Policy 03, Lake Como  
22

Thank you!

Questions?

Remarks?

Comments?

Advises?

Thursday 5th June 2003  
Policy 03, Lake Como  
23