Interactivity for Reactive Access Control
To appear in Secrypt 2008

Yehia ElRakaiby, Frederic Cuppens & Nora Cuppens Boulahia
TELECOM Institute ; TELECOM Bretagne
Département Réseau Sécurité et Multimédia - RSM Department

10 Juin 2008
Outline of Topics

Interactivity for Reactive Access Control

- Introduction & Motivation
- Overview
- Basic Concepts
- Formal Model
- Policy Enforcement & Interpretation
- Application Example
- Conclusion
Introduction

Evolution of the Computing & communication capabilities of networks and electronic devices

- New Intelligent Context-aware Environments

Figure: Example SIP PIDF Presence Information
Motivation

Current Access Control Systems

- Passive Systems, \( e.g. \ RBAC_0 \)
  \( \text{Role} \times \text{Permission} \)

- Dynamic Systems, \( e.g. \ OrBAC, \ GRBAC \)
  \( \text{Role} \times \text{Permission} \times \text{Context} \)

Characteristics

- Anticipative models as all rules have to be predefined for every possible access request
Interactivity for Access Control

Specification of the Access Policy at the Time of the Request

- Permit the active participation of a third party in the evaluation of security policies e.g. A patient’s file on some hospital’s database \((Role \times Permission \times Context \times Patient)\)

- Handle Unexpected Situations e.g. Unexpected absences due to illness. \((Role \times Permission \times Context \times DepartementHead)\)

- Awareness of Important Accesses

- Just In-time Specification of Access Control Policies & Per-Access if Needed e.g. Access to Files of Ongoing Projects, Access to PCs in an Internet Cafe \((Role \times Permission \times Context \times Admin)\)

- Policy Retrieval from another Policy Decision Point \((Role \times Permission \times Context \times Server_1)\)
System Overview

Two Rule Specification Schemes

- In advance
- At the time of the request

Figure: System Operation Overview
**OrBAC Policies & Contexts**

**OrBAC Policies**
- Contextual Model
- Rules → Organization

**Context Representation**
- Separation context/security rule
- Representation: $Hold(S, A, R, Context)$
  
  $Hold(S, A, R, childAtSchool) \leftarrow Attribute(age, S, X), X < 10,$
  
  $Attribute(location, S, school)$
  
  $Hold(S, A, R, morning) \leftarrow after \_time(08 : 00), before \_time(12 : 00)$

- **OrBAC Context Language Supports the AND, OR and NOT operators:**
  
  $Permission(Students, EnterPlayground, childAtSchool & morning)$
Object Organization

Organizational Entities

- Policies are defined over the organizational entities Role, Activity and Views
- Easy Object Manipulation is Desirable
  - Reduction of Policy Definition & Deployment Time
- Linking activities and views
  - Logically interconnect activities and views by associating to every resource/view an activity containing all the operations it supports
- Every resource in the model is associated to one manager
Example

Organizing Objects
- Views $\subseteq 2^{\text{Resources}}$
- Activities $\leftarrow$ Objects/Views
- Sub-Activities $\subseteq$ Activities
- Define Permissions on Activities
  $\text{Permission(} \text{Family, classicalCDs} \text{)}$
  $\text{Permission(} \text{Family, readOnlyRock} \text{)}$

Figure: Object Organization Example
Formal Model

Basic Elements
- Subjects ($S$), Resources ($R$), resource-Types ($T$), Actions ($A$), Operations ($O$), Attributes ($Att$) and Contexts ($C$)
- Dynamic Context ($C_d$) is of type boolean

Organizational Entities
- Roles ($\mathcal{R}$), Views ($\mathcal{V}$), Activities ($\mathcal{A}$)

Policy Elements
- $P \subseteq \mathcal{R} \times \mathcal{A} \times C \times C_d$
  
  Ex: $P(\text{family}, \text{rockCDs}, \text{atHome}, \text{true})$

System Messages
- Access-request (AR): $AR \subseteq S \times \mathcal{A}$
- Grant (GR): $GR \subseteq S \times O$
- System-Request Messages (SR): $SR \subseteq S \times S \times \mathcal{A} \times ID$
- Manager-response Messages (MR): $MR \subseteq S \times \mathcal{A} \times C \times ID$
Policy Interpretation using Active Rule

An Active Rule

- **on** event **if** condition **then** action

Enforcing the system’s policy

- 2 input messages (AR)-(MR)
- 3 output messages (GR)-(DN)-(SR)

**on** Reception of Message **if** conditions **then** Sending of Message

Example: The Access-Request/Grant Rule:

```plaintext
on AR(S₁, A₁)
if P(R₂, A₂, Context, false),
DerivedMember(S₁, R₂),
Compatible(A₁, A₂),
DerivedMember(Operation(R, A), A₁),
Hold(S₁, R, A, Context)
then Grant(S₁, Operation(R, A))
```
Policy Interpretation using Active Rule

Conflict Resolution

- Contextual/dynamic permission conflict
- Resolved by prioritizing dynamic permissions

Timeout Situations

- $C_d \subseteq D \times DA$
  
  Where $DA \in \{accept, deny, other\}$

  Ex: **on** timeOut($id$)
  
  **if** Interaction($S_1, A_1, C_d(D, DA), id$), $DA = deny$
  
  **then** Deny($S_1, A_1$)
Example Policy

Consider the following policy

- $P_1$: $P(\text{family}, \text{classicalCDs}, \text{default}, \text{false})$
- $P_2$: $P(\text{family}, \text{rockCDs}, \text{jackAvailable}, \text{dc}(60, \text{other}))$
  The context $\text{jackAvailable}$ is defined as:
  $C_1$: $\text{Hold}(S, R, A, \text{jackAvailable}) \leftarrow \text{Attribute}(\text{status}, \text{jack}, \text{available})$
- $P_3$: $P(\text{family}, \text{onlyReadRockCDs}, \text{atHome}, \text{false})$
  The context $\text{atHome}$ is defined as:
  $C_2$: $\text{Hold}(S, R, A, \text{atHome}) \leftarrow \text{Attribute}(\text{location}, S, \text{home})$
Example Scenario

Consider the following request

- $AR(tom, rockCDs)$

The resource manager can

- Limit the authorized operations
  $MR(tom, readOnlyRockCDs, default, id)$
- Deny the access
  $MR(tom, rockCDs, false, id)$
- Require the verification of some context
  $MR(tom, rockCDs, janeNotAtHome, id)$
  $Hold(S, R, A, janeNotAtHome) \leftarrow \neg Attribute(location, jane, atHome)$
- Timeout: only operations defined in $readOnlyCds$ are allowed
Conclusion

We have discussed the Advantages of Interactivity for Access Control

- Awareness
- Handling Unexpected Situations

We have proposed a formal model that extends context-aware models to handle interaction

We have shown how the policy can be enforced using ECA rules

We have proposed an intuitive object organization scheme
Future Work

Usage Control

- Adding ongoing controls to the model
- Just-in-time delegation of capabilities
- Contacting several subject
Thank you for your attention...