Project AutoMate
SESAME: Dynamic Context Aware Access Control

G. Zhang, The AutoMate Group
The Applied Software Systems Laboratory
Rutgers, The State University of New Jersey
http://automate.rutgers.edu

CAIP Autonomic Computing Tutorial/Workshop
June, 2003

Overview

• Security Issues in Autonomic Computing
• SESAME – Access Control Mechanism for AutoMate
• RBAC Introduction
• DRBAC Model
• DRBAC Model Explanation
• SESAME Architecture
• A Prototype Implementation in Discover
• Current Issues

Security Issues in Autonomic Computing

• Authentication
• Authorization, Access Control
• Intrusion Detection
• Security Policy Definition and Reasoning
• Resistant to Fraud and Persuasion
• Privacy
• Digital Signature, Non-repudiation
  – Crucial for E-commerce Application
Authorization, Access Control

- The environment will be heterogeneous and dynamic
- Components amount will be huge and across domain
- Centralized Authorization is not sufficient
- Global name space has constraints.
- Access Control should be Context Aware
- DAC, MAC, RBAC
- Fine grained access control mechanism
  - Our approach – SESAME (Environment Sensitive Access Management Engine)
  - Dynamic Role Based Access Control

RBAC Introduction

- Alternative to traditional discretionary access control (DAC) and mandatory access control (MAC)
- In RBAC, users are assigned roles and roles are assigned permissions.
  - RBAC0: the basic model where users are associated with roles and roles are associated with permissions.
  - RBAC1: RBAC0 with role hierarchies.
  - RBAC2: RBAC1 with constraints on user/role, role/role, and/or role/permission association.
- Cost of administrating RBAC is proportional to U+P while the cost of associating users directly with permissions is proportional to U*P

SESAME-DRBAC Model

- Current access control mechanism focus on relatively static scenarios where access depends on identity of the subject.
- Autonomic Computing – Self Protecting (Context aware, Dynamic)
- Access capabilities and privileges of a component not only depend on its identity but also on its current context (i.e. current time, location, system resources, network state, etc.) and state.
- Extension of RBAC (context information play a role in access decision)
SESAME-DRBAC Model Explanation

- Central Authority (CA) maintains the overall role hierarchy for each domain.
- Each entity is assigned a subset of the role hierarchy.
- Context agent monitors the context for the Entity and dynamically changes the active role (Role State Machine).
- Context agent at the subject resource will use environment and state information to dynamically adjust the permissions for each role (Permission State Machine).

Role & Permission State Machine

- Role Hierarchy
- Permission Hierarchy
A Prototype-DRBAC in Discover

- Discover enables geographically distributed scientists and engineers to collaboratively access, monitor and control applications, services, resources and data on the Grid using pervasive portals.
  - Discover Collaborative Portals
  - Discover Middleware Substrate
  - DIOS Interactive Object Framework (DIOS)
Role & Permission Hierarchy in Discover

<table>
<thead>
<tr>
<th>Roles</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super User</td>
<td>P1, P2, P3</td>
</tr>
<tr>
<td>Basic User</td>
<td>P2, P3</td>
</tr>
<tr>
<td>Guest</td>
<td>P3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>View object, View object, Basic</td>
</tr>
<tr>
<td>P2</td>
<td>View object, Basic</td>
</tr>
<tr>
<td>P3</td>
<td>Basic</td>
</tr>
</tbody>
</table>

Permission Hierarchy of One Application

Access Control Policy – An Example

```
<ROLE_TRANSITION>
<POLICY>
  <SUBJECT>Engelbart, SUSAN</SUBJECT>
  <BEGN_ROLE>Super User</BEGN_ROLE>
  <EVENT>Unsecure Link/HOLE</EVENT>
  <END_ROLE>Basic User</END_ROLE>
  <POLICY>
    ...<POLICY>
  <END_ROLE>
  <ROLE_TRANSITION>
```
A Prototype-DRBAC in Discover

Current Issues

• Must guarantee the security of the context information.

• The active role of the user and the active permission of the role will change dynamically. We need some mechanism to keep the consistency.

• Combine with available authentication mechanism.

• Delegation with DRBAC

Bibliography


