Dynamic Context-aware Access Control for Grid Applications

Guangsen Zhang and Manish Parashar
The Applied Software Systems Laboratory
Rutgers, The State University of New Jersey

4th International Workshop on Grid Computing (Grid2003)
November, 2003
Outline

- Motivation
- Related work
- SESAME – Dynamic Context Aware Access Control Mechanism
- RBAC introduction
- DRBAC model
- DRBAC model explanation
- A Prototype implementation in Discover
- Current issues
Motivation

- Pervasive Grid applications
- Better security assurance
- Context information play a role
- Access capabilities and privileges will be difference
  According to context information
- Dynamic and heterogeneous Grid computing environment
Related Work

• GSI (Grid Security Infrastructure)
  – *Has been accepted as the primary authentication mechanism for the Grid. (Globus Project)*

• Akenti
  – *User-conditions certificate and attribute certificate, grant access by matching of these two certificates*

• CAS (Community Authorization Service)
  – *Resource providers grant access to community accounts as a whole*

• Authorization service for ad-hoc collaboration
• Authorization service for component-based Grid applications
Why SESAME?

- Existing access control mechanisms do not address access control issues for pervasive applications.
- The environment will be heterogeneous and 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.
- Fine grained access control mechanism
  - Our approach – SESAME (Scalable Environment Sensitive Access Management Engine)
    - Dynamic Role Based Access Control
    - Extension of RBAC (context information play a role in access decision)
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
Model Definition

- **USERS.**
  - A set of entities whose accesses are being controlled.

- **ROLES.**
  - A role is a job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role. ROLES represents a set of roles.

- **PERMS.**
  - A permission is an approval to access one or more DRBAC protected resources. PERMS represents a set of permissions.

- **ENVS.**
  - ENVS represents the set of context information for the system.

- **UA. (User Assignment)**
  - The mapping that assigns a role to a user.

- **PA. (Permission Assignment)**
  - The mapping that assigns permissions to a role.
Model Definition

- USERS, ROLES, PERMS, ENVS and SESSIONS
- ACT ROLE and ACT PERMISSION
- \( UA \subseteq \text{USERS} \times \text{ROLES} \)
- \( PA \subseteq \text{PERMS} \times \text{ROLES} \)
- Assigned roles\((u: \text{USERS}, e: \text{ENVS}) \rightarrow 2^{\text{ROLES}}\)
- Assigned permissions\((r: \text{ROLES}, e: \text{ENVS}) \rightarrow 2^{\text{PERMS}}\)
- \( RH \subseteq \text{ROLES} \times \text{ROLES} \)
- \( PH \subseteq \text{PERMS} \times \text{PERMS} \)
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)
Discover Architecture
A Prototype-DRBAC in Discover

Diagram showing the components and interactions of a prototype-DRBAC system in Discover, including role state machine, context agent, authentication and authorization service, policy repository, middleware substrate, and DIOS enabled applications.
Access Control Policy – An Example

<ROLE_TRANSITION>
  <POLICY>
    <SUBJECTID>gszhang</SUBJECTID>
    <BEGIN_ROLE>Super User</BEGIN_ROLE>
    <EVENT>Unsecure Link</EVENT>
    <END_ROLE>Basic User</END_ROLE>
  </POLICY>
</ROLE_TRANSITION>
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>Steer 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

Super User's Permission

Basic User's Permission

Guest's Permission

Grid 2003, November 2003
A Prototype-DRBAC in Discover
Experimental Evaluation

Table 3. Interaction time in ms. for different context event frequencies.

<table>
<thead>
<tr>
<th>Event frequency</th>
<th>Time (ms.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>2300</td>
</tr>
<tr>
<td>1min</td>
<td>4732</td>
</tr>
<tr>
<td>2min</td>
<td>4403</td>
</tr>
<tr>
<td>3min</td>
<td>4102</td>
</tr>
<tr>
<td>4min</td>
<td>3482</td>
</tr>
<tr>
<td>5min</td>
<td>3104</td>
</tr>
</tbody>
</table>

Table 4. Interaction time in ms. for different number of roles.

<table>
<thead>
<tr>
<th>Number of Roles</th>
<th>Time (ms.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>2300</td>
</tr>
<tr>
<td>5</td>
<td>2520</td>
</tr>
<tr>
<td>6</td>
<td>2608</td>
</tr>
<tr>
<td>7</td>
<td>2804</td>
</tr>
<tr>
<td>8</td>
<td>2920</td>
</tr>
<tr>
<td>9</td>
<td>3004</td>
</tr>
</tbody>
</table>

Table 5. Interaction time in ms. for different number of permissions.

<table>
<thead>
<tr>
<th>Number of Permissions</th>
<th>Time (ms.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>2300</td>
</tr>
<tr>
<td>5</td>
<td>2500</td>
</tr>
<tr>
<td>6</td>
<td>2602</td>
</tr>
<tr>
<td>7</td>
<td>2698</td>
</tr>
<tr>
<td>8</td>
<td>2804</td>
</tr>
<tr>
<td>9</td>
<td>2912</td>
</tr>
</tbody>
</table>
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
Conclusion

- SESAME complements current authorization mechanisms
- Dynamic grant permissions based on context
- Overhead for pervasive Grid applications is reasonable
- Can be used to enhance the security of the Grid applications.
QUESTIONS?