A Question...

- Computer security beyond OSes
  - The advent of the Internet, e-business
  - Network security solution (SSL)
  - Public key infrastructure
  - Access control
- So, what was the biggest challenge in enterprise-wise access control management?

Another Question

Then, what is the difference between roles and other intermediaries such as security labels?

Role-based Access Control (RBAC)

RBAC

- A mechanism which allows and promotes an organization-specific AC policy based on roles
- Enabling to support the development and adoption of new complex networked security systems with significant contribution
  - The net benefits through 2006 are $671.1 million [NIST Planning Report 02-01, 2002]
- Has become widely accepted as the proven technology
  - Many organizations based access control decisions on the roles that individual users take on as part of organizations

Why RBAC

- Reasons to use roles
  - Expressive power
  - Stability
- RBAC security principles
  - Least privilege
  - Separation of duties
  - Abstract operations
RBAC Models

- RBAC96 Family
  - RBAC0: Basic Model
  - RBAC1: Hierarchical Roles
  - RBAC2: Constrained Roles
  - RBAC3: RBAC1 + RBAC2

Why Roles

- Fewer relationships to manage
  - From O(mn) to O(m+n), where m is the number of users and n is the number of permissions
- Roles add a useful level of indirection

RBAC0: Formal Expressions

- U, R, P, S (users, roles, permissions, and sessions)
- UA ⊆ U × R (user assignment)
- PA ⊆ P × R (permission assignment)
- user: S → U
- roles: S → 2^R
  - Requires roles(s) ⊆ { r | (user(s), r) ∈ UA }
**RBAC1: Formal Expressions**

- U, R, P, S, UA, PA, and user unchanged from RBAC0
- RH ⊆ R × R: a partial order on R, written as ≥
- roles: S → 2^R
  - Requires roles(s) ⊆ { r | ∃ r' ≥ r | (user(s), r') ∈ UA }

**Role Hierarchies**

- User inheritance
  - r1 ≥ r2 means every user that is a member of r1 is also a member of r2
- Permission inheritance
  - r1 ≥ r2 means every permission that is authorized for r2 is also authorized for r1
- Activation inheritance
  - r1 ≥ r2 means that activating r1 will also activate r2

**RBAC2**

- No formal model specified
- Constraints are added
  - Mutually exclusive roles
  - Cardinality
- Constraints are for laying out higher level organization policy

**RBAC3**

- No formal model specified
- Constraints are added

- Users are
  - Human beings or
  - Other active agents
- Each individual should be known as exactly one user
**Roles as Policy**

- A role brings together
  - A collection of users and
  - A collection of permissions
- These collections will vary over time
  - A role has significance and meaning beyond the particular users and permissions brought together at any moment

**A Question?**

- What is the difference between groups and roles?
  - A group often defined as
    - A collection of users
  - A role
    - A collection of users and
    - A collection of permissions
- Some authors define role as
  - A collection of permissions

**Hierarchical Roles**

- Primitive permissions
  - read, write, append, execute, etc
- Abstract permissions
  - credit, debit, inquiry

**Permissions**

- Permissions are positive
- No negative permissions or denials
  - Negative permissions and denials can be handled by constraints

**User-Role Assignment**

- A user can be a member of many roles
- Each role can have many users as members
- URA97 model
**Implicit User Assignment**

- A permission can be assigned to many roles
- Each role can have many permissions

**Explicit User Assignment**

- A user can invoke multiple sessions
- In each session a user can invoke any subset of roles that the user is a member of

**Permission-Role Assignment**

- A permission can be assigned to many roles
- Each role can have many permissions

**Sessions**

- A user can invoke multiple sessions
- In each session a user can invoke any subset of roles that the user is a member of

**Constraints**

- Applied to all components in RBAC
- Example: mutually exclusive roles
  - Static exclusion: the same individual can never hold both roles
  - Dynamic exclusion: the same individual can never hold both roles in the same context

**Separation of Duty**

- Purchasing Manager
- Accounting Payable Manager
Expressive Power

- Can be configured to do MAC
  - Roles simulate clearances
- Can be configured to do DAC
  - Roles simulate identity

Goal

- Decentralize the administration of RBAC, i.e., allowing others to change parts of (UA, PA, RH)

Overview

- There exist a set of administrative roles that are disjoint from the regular roles

ARBAC97 Model for Role-based Administration of Roles

Role Hierarchies

- Prerequisite condition
  - e.g., r1\lor (r2\land \neg r3) is such a condition
- can_assign
  - e.g., can_assign(a, cond, \{r4,r5,r6\})
- can_revoke
  - e.g., can_revoke(a, \{r4,r5\})

URA97 Grant Model
URA97 Grant Model

- can-assign

<table>
<thead>
<tr>
<th>ARole</th>
<th>Prereq Role</th>
<th>Role Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO1</td>
<td>ED</td>
<td>[E1,PL1]</td>
</tr>
<tr>
<td>PSO2</td>
<td>ED</td>
<td>[E2,PL2]</td>
</tr>
<tr>
<td>DSO</td>
<td>ED</td>
<td>(ED,DIR)</td>
</tr>
<tr>
<td>SSO</td>
<td>E</td>
<td>[ED,ED]</td>
</tr>
<tr>
<td>SSO</td>
<td>ED</td>
<td>(ED,DIR)</td>
</tr>
</tbody>
</table>

URA97 Revoke Model

- can-revoke

<table>
<thead>
<tr>
<th>ARole</th>
<th>Role Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO1</td>
<td>[E1,PL1]</td>
</tr>
<tr>
<td>PSO2</td>
<td>[E2,PL2]</td>
</tr>
<tr>
<td>DSO</td>
<td>(ED,DIR)</td>
</tr>
<tr>
<td>SSO</td>
<td>[ED,DIR]</td>
</tr>
</tbody>
</table>

URA97 Revoke Model

- Strong revocation
  - Revokes explicit membership in a role and its seniors
  - Authorized only if corresponding weak revokes are authorized
- Weak revocation
  - Revokes explicit membership in a role

PRA97

- Treat permission assignments as dual to user assignment
  - can_assign
    - e.g., can_assign(a, cond, [R4,R5,R6])
  - can_revoke
    - e.g., can_revoke(a, [R4,R5])

Role Administration (RA)

- Definition
  - RA is administrative routine works for implemented RBAC.
- Goal
  - RA is to manage and maintain the designed and implemented roles. Administration of roles should be done cautiously so as not to diverge from organizational security policies.
- Functionality
  - RA enables roles and role hierarchies to be managed properly in conjunction with users. It involves user assignment, constraints present in user assignment, in addition to the other components in implemented RBAC.

RolePartner: A Role Administration Tool
RA Methodology

- Reorganization of RBAC components into three management constituents in order to reflect structural and behavioral characteristics in role administration

Structural components
- Represent both semantics and syntaxes of U, R, P, RH, D, Op, and C.

Functional components
- Represent UA and PA, in addition to USERS and RULES.

Informational components
- Represent repositories such as the LDAP directory

RolePartner: Design

- Model-driven
  - RBAC96 reference models
  - ARBAC97
- Component-based
  - Components in the models are mapped to RolePartner components
- Design procedures
  - Requirement specification
  - Component specification
    - Structural overview
    - Operational architecture
    - Database design

Design Goals

User-Friendly GUI
- User-centric view for easily managing associated users and permissions.

Complete Support for RBAC Functionalities
- Able to build full RBAC structural and functional components.

Easy-to-Use

Standardization

Availability

Applicability

Available in different system environments
- Platform-independence of, the support of various formats of authorization policies in, and scalability of the system

Applicable in different organizational policy environments
- Flexible enough to be configured and customized

Functional Requirement

- Add/delete/display roles
- Add/delete/display permissions
- Add/delete/display in role
- Set/drop constraints
- Function Roles
- Add/delete/display users
- Change users to roles
- Chat drop constraints
- Function Users

Structural Overview

User Interface (GUI)

Executive Services (ES)

RBAC Structural Service (RSS)

RBAC Functional Service (RFS)

RBAC Constraint Service (RCS)

Data Encoding Service (DES)

Data Synchronization Service (DSS)

Network Interface (NI)

Operational Architecture
Sequence Diagram

Directory service (Role DB)

RolePartner

Constraint Designer

Deployment Models

Role Engineering
**Engineering approach**

**Role Engineering Approach**

**Engineering problem**

Analyzing, mapping, and refining semantics of purchasing manager role

Information Security Group

• Separation of duties, lease privilege, ...

• Employee data, screening, ...

• e-Procurement system, ERP, ...

Purchasing manager role

HR department

System Administrator Group

**Engineering schemes**

- Top-down
- Bottom-up
- Hybrid

- Each has its own advantages and pitfalls depending upon the varying contexts.
  - Top-down: usually ignore existing permissions
  - Bottom-up: possibly ignore business functions within an organization

**Top-down scheme**

- Scheme where permissions are derived from roles
  - Use of abstract concepts such as work-pattern and business processes
  - The abstract concepts are analyzed and decomposed into functionally smaller units

**Bottom-up scheme**

- Scheme where permissions works as building blocks to compose roles
  - Use of abstraction such as scenarios and business functions, or certain attributes of objects or operations

**Role mining**

Users

Permissions

Roles

Permissions
Role mining

Project Guideline: An Example

Typical Approach
- Environment/Domain
- Access Control Requirements
- Access Control Model
- System Architecture
- Implementation through Mechanisms

Agenda
- Discuss how “ERMPAM” can be materialized
  - E: Distributed environment
  - R: Access control issues
  - M: Role-based access control
  - A: Privilege Management Infrastructure
  - M: Feasibility test through COTS technologies

Environment - EAM

Environment
Discuss how “ERMPAM” can be materialized

E: Distributed environment
R: Access control issues
M: Role-based access control
A: Privilege Management Infrastructure
M: Feasibility test through COTS technologies

From closed to open environment
- The Internet technologies have been replacing traditional business processes with web-based business solutions

Administrative complexity and blurring boundaries
- Recent trends in web-based solutions increase the administrative complexity of access management and blurs enterprise boundaries.

Lack of support for interoperability
- EAM was considered as a possible solution, but its authorization still needs to be investigated, especially in terms of interoperability.

Centralized administration
- All application servers are dependant on Access Control (AC) Server
- AC Server specifies roles and its permissions

Coherent access control policy
- Role-specification and role-assignment policies are managed in AC Server
- A change of access control policy in one place can propagate through distributed components in the domain
- One Access Control Policy in AC Server will govern the entire domain

AC Server – Trusted Third Party
- Management of trust relationship is easier and more secure

Integrity of User Information
- Sensitive user’s personal information is restrained in AC Server
- Registration in one place and access everywhere will provide integrity of user information

Users’ Privacy
- Least users information required for access control is revealed to application server

Need to make access control decisions on “the roles that individual users take on as part of the organization”

Need to centrally control and manage access rights based on the organization’s security policy

Need to provide application independent access control policy

Agenda
- Discuss how “ERMPAM” can be materialized
  - E: Distributed environment
  - R: Access control issues
  - M: Role-based access control
  - A: Privilege Management Infrastructure
  - M: Feasibility test through COTS technologies
### Why RBAC

- **Least privilege**
  - Only those permissions required for the tasks performed by the user in the role are assigned to the role
- **Separation of duties**
  - Invocation of mutually exclusive roles can be required to complete a sensitive task
- **Data abstraction**
  - Instead of the read, write, execute permissions typically provided by the operating system, abstract permissions can be established

### Agenda

- Discuss how “ERMPAM” can be materialized
  - E: Distributed environment
  - R: Access control issues
  - M: Role-based access control
  - A: Privilege Management Infrastructure
  - M: Feasibility test through COTS technologies

### PMI MODELS

**Control Model**
- Environmental Variables

**Object Method**
- Roles Model

**Roles Model**
- Role assignment

**Delegation Model**
- Delegation of role

**PMI MODELS**

**Optional Model**
- Object Method

**General Model**
- Role assignment

### Attribute Certificate (AC)

<table>
<thead>
<tr>
<th>Public Key Certificate (PKC)</th>
<th>Attribute Certificate (AC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Version</td>
</tr>
<tr>
<td>Serial Number</td>
<td>Serial Number</td>
</tr>
<tr>
<td>Signature ID</td>
<td>Signature ID</td>
</tr>
<tr>
<td>Subject</td>
<td>Holder</td>
</tr>
<tr>
<td>Issuer</td>
<td>Issuer</td>
</tr>
<tr>
<td>Validity Period</td>
<td>Validity Period</td>
</tr>
<tr>
<td>Subject Public Key Info</td>
<td>Attributes</td>
</tr>
<tr>
<td>Extensions</td>
<td>Extensions</td>
</tr>
<tr>
<td>Algorithm ID</td>
<td>Algorithm ID</td>
</tr>
<tr>
<td>Signature</td>
<td>Signature</td>
</tr>
</tbody>
</table>

### Why PMI?

- Essential requirement is authorization not authentication
  - X.509 Public Key Certificate provides authentication service based on PKI
  - More important to know what a user can do than who a user is
- Difficult to manage privilege information in PKI
  - Complicated issuing process including user identification
  - In general validity of privilege is much shorter
- Identity certificate is passport and AC is visa
  - Need to integrate PKI with PMI

### How

1. Need two different attribute certificates based upon PMI roles model.
   - Role assignment attribute certificate (RAAC): assigning roles to a user
   - Role specification attribute certificate (RSAC): assign permissions to a role
2. Need to identify and design role administration components to manage RBAC policies in a systematic and efficient manner.
3. Need to investigate possible system architectures so that we can fully utilize attribute certificate.
Two Attribute Certificates

<table>
<thead>
<tr>
<th>Role Assignment AC</th>
<th>Role Specification AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Version</td>
</tr>
<tr>
<td>Serial Number</td>
<td>Serial Number</td>
</tr>
<tr>
<td>Signature ID</td>
<td>Signature ID</td>
</tr>
<tr>
<td>Issuer</td>
<td>Issuer</td>
</tr>
<tr>
<td>Validity Period</td>
<td>Validity Period</td>
</tr>
<tr>
<td>Roles</td>
<td>Roles</td>
</tr>
<tr>
<td>Permissions</td>
<td>Permissions</td>
</tr>
<tr>
<td>Signature</td>
<td>Signature</td>
</tr>
</tbody>
</table>

System Architecture

Push Model

- Server: Enhanced Performance (+)
- AC Management in client (-)
- Complicated Client-Server Protocol (-)

Pull Model

- Increase Server Overhead (-)
- Simple client module (+)
- Simple modification in existing Client-Server Protocol (+)

Role Specification

- A set of roles and permissions is defined according to Rule-Based Access Control Policy
- Application Server requests an AC with a predefined role
- AC Server issues the AC that contains the predefined role
- Centralized Access Control Policy Administration in a distributed environment
- RBAC Policy is stored in LDAP or internal database

Role Assignment

- A client is assigned to a specific role
- When a client requests an AC, AC Server issues a role-assignment AC
- RAAC contains the assigned role that a client can use to access an application server
Decision Making Engine

Access Request

Access Control Policy Server

Access Granted

Web Application Server

Decision

Decision Request

Decision Response

Role-Based Access Control Engine

Decision Making Engine

Role Administration Tool

X.509 AC Manager

Conclusions

- Showed how distributed networked environments can support scalable access control policies
- First attempt to accommodate PMI roles model
- Introduced systematic role engineering and system architecture
- Demonstrated the feasibility of proposed approach through a proof-of-concept prototype implementation

Agenda

- Discuss how “ERMPAM” can be materialized
  - E: Distributed environment
  - R: Access control issues
  - M: Role-based access control
  - P: Attribute certificate
  - A: Pull and Push architecture
  - M: Feasibility test through COTS technologies

Role Administration Tool

X.509 AC Manager