UC Irvine Access Management Use Cases

The most up to date version of this document can be found at https://services.adcom.uci.edu/wiki/display/public/Enterprise+Authorization+Use+Case+Document

As of May 22nd, 2009, this is our use case document

Version 0.1

Table of Contents

Glossary

The goal was to stay as close as possible to the MACE-paccman-glossary.

Actor Name	Description
Auditor	External entity that audits privileges
DSA (Grantor)	A principal authorized to delegate some authority. Subjects are assigned to DSAs to manage the privileges of the subject.
Super DSA (Super Grantor?)	A principal authorized to grant DSAs the privilege of granting a set privileges. Super DSA delegates authorization decisions to the DSA. Super DSAs also administer the various privileges.

Use Cases

0. Define Privileges

Actors	Super DSA
Goal	To establish privileges
Summary	Super DSAs need to define the different privileges and the scopes associated with them. The scope in which the privilege is used determines the fine grained access. For example, a subject is allowed to make purchases only for departments x, y, and z. In most cases, the scope refers to the campus hierarchy. The service needs to be flexible enough to allow us to define multiple hierarchies. Sometimes, wildcards are used to denote that the scope applies to all units at a certain level in the hierarchy. A group of asterisks can be thought of as a single wildcard value. No support is needed for values like "6*33**" e.g. 1. Location *, Org 1000, Div 1500, Subdiv **, Department ****, Account ***** 2. Control Point 02, Home Dept 317500 3. Primary Unit 430000, Secondary Unit ****** Also, some cases require a "workflow" system that has hooks to external systems (e.g. training is required, central office approval required). The granularity needs to be flexible so that many different "types" of scopes can be handled. e.g. 4. User X can make purchases less than $5,000 5. User Y can access the system between 10-11 AM 6. User X is allowed to view pages X, Y, Z 7. User X is allowed to edit pages X, Y, Z

Primary Path	1. Super DSA authenticates 2. Super DSA creates a new privilege 3. Super DSA assigns a scope type (among the types listed above) 4. A workflow determines which privileges are in conflict with the new privilege by using the principle of separation of duties

1. Delegated Authorization

Actors	Super DSA, future DSA, stakeholder
Goal	The ability to assign a subject to grant access for a subset of privileges.
Summary	A Super DSA defines who can act as a DSA to assign privileges. The privileges that a DSA can assign are bounded by what the Super DSA grants to the DSA.
Precondition	• The privileges to be delegated have been created (See use case 0)
Primary Path	1. Subject submits a request become a DSA 2. Stakeholder approves the request 3. Super DSA authenticates 4. Super DSA assigns a subject to be a DSA 5. Super DSA defines which privileges the DSA is allowed to grant
Postcondition	• The subject is able grant privileges (becomes a DSA)

2. Standard API for Determining Access

Actors	External Application, end users
Goal	To provide a standardized way to determine access.
Summary	A well documented, standard (SAML, XACML, SOAP/REST, etc) way of determining if a subject is authorized to perform a business function. The API should be available across all platforms. This is essential not only to consolidate the authorization service for homegrown applications but also vendor or otherwise black box applications.
Precondition	• The external application is authorized to make requests
Primary Path	1. The External Application asks the central server whether subject X currently is granted privilege Y limited by scope Z 2. The central service responds with a yes/no
2a) Alternate Path (Get Authorized Scope)	1. The external application asks the central server for what scope does subject X have for privilege Y 2. The central service returns the scope
2b) Alternate Path (Find Valid Subjects)	1. The external application asks what subjects have been granted a certain privilege given action within the provided scope 2. The central service returns a list of subjects with privileges
2c) Alternate Path (Get Subject's Access)	1. The external application asks which privileges does subject X have 2. The central service responds with a list of current privileges and optionally the scope of each privilege

3. Standard API for Granting Access

Actors	External Application, DSAs, end users
Goal	To provide a mechanism to allow applications other than the central GUI to provision access.
Summary	In addition to determining access within applications, there is a desire to manage authorization in the applications themselves. e.g. Students can decide what their parents can do on their behalf or what information the student would like to release to others. Another use for this feature would give DSAs the ability to provision access withing their own applications without being tied to the central product's GUI.
Primary Path	1. End user/DSA authenticates to an application 2. The user is presented with a GUI that helps determine the access that the user is granting to others 3. The application makes a call to the central system

Post Conditions	• If no error occurs, the access is granted to the "other" user(s)

4. Assign Privileges With Low Level Granularity

Actors	DSA, Subject
Goal	Assign a privilege to a subject with additional parameters.
Summary	In most cases, having a privilege is not sufficient to grant access for a business function.
Precondition	• The subject has requested access • The request has been approved by a stakeholder • The DSA assigned to a user is able to grant the permission in question
Primary Path	1. DSA assigns a privilege to a subject 2. DSA defines the scope of the privilege. 3. The scope that is assigned to the privilege passes business validation (such as a valid hierarchy define in Use Case 0)
Alternate Path (Workflow Requirement)	1. DSA assigns a privilege to a subject 2. DSA defines the scope of the privilege 3. External workflows complete, effectively approving or vetoing access 4. Access to a resource is determined by the outcome of the workflow
Alternate Path (No hierarchy)	1. DSA assigns a privilege to a subject 2. (Optionally) External workflows complete, effectively approving or vetoing access
Post Conditions	• User is granted the privilege limited by the scope • The privilege that has been granted must adhere to the separation of duties principle

5. Assign Privileges for Files / Directories Supported by Servers (CMS)

Actors	DSA
Goal	Restrict the viewing of individual static files/folders to authorized users only.
Summary	Static resources need to be restricted as well. Our CMS pushes static content to servers and there needs to be a way to push privileges to these servers as well.
Preconditions	• The subject has requested access • The request has been approved by a stakeholder
Precondition	• A privilege has already been defined that represents access to static resources
Primary Path	1. DSA assigns privileges to a subject 2. The authorization model is pushed to the various web/application/ldap servers or local agent is deployed in front of web server that can talk to authorization server and handle access

6. Assign Subjects to Groups (Coarse-Grained Control)

A group is a subject

Group membership is maintained independently of privileges. Groups can either be based on attributes ("Subject A is an employee") or roles ("Subject A can hire future employees")

Actors	DSA, Super DSA, Subject
Goal	Provide RBAC and group-based privileges.
Summary	Assigning users to groups allows DSAs to grant privileges to a group of users while keeping the management of the groups external to the privilege management. Groups can be modeled after roles.

Primary Path	1. Super DSA grants a DSA the privilege to modify the membership of a group 2. The DSA adds/removes subjects to/from a group 3. Workflows, if applicable, complete 4. Membership is granted or vetoed
Alternate Path (Self-Subscribed)	1. Subjects apply to be a member of a group 2. Workflows, if applicable, complete 3. Membership is granted or vetoed
Alternate Path (Entitlements)	1. Based on business attributes such as title, a subject is automatically included in a group

7. Audit Privileges

Actors	Auditor
Goal	Provide a mechanism to audit privileges.
Summary	A mechanism is required that allows an auditor to query a subject's access on at any given time. In other words, given a user and a time frame what privileges were assigned to said user. Another case is finding privileges in conflict with business policies and auditing/enforcing separation of duties. Auditors also want to know who requested the access and why, and who granted the access and relevant workflow approvals.
Primary Path	1. Auditor enters the user id and time range to search on 2. System provides a list of privileges for the given user in the given time range
Alternate Path (Who/Why)	1. The auditor searches for a user 2. The auditor is presented with a list of current and former privileges 3. The auditor is able to drill down and see who/what/why information

8. Provision Privileges to External System

Actors	DSA
Goal	Provision access control to the OS
Summary	It is desirable to control access to various resources such as Active Directory, OS, external database, web service, JMS, Email etc.
Primary Path	1. DSA assigns a privilege determining access to network resources 2. The system pushes these privileges to the underlying mechanisms, in turn granting and revoking access to individual resources 3. The underlying mechanisms determine if a subject can access the resource

Functional Requirements

Campus IdM Integration

The solution must integrate with the campus single sign-on, UC wide federated authentication, and other existing IdM systems.

Third Party Access

Access control solution should handle access to applications shared with other non-UCI users as well as third-party affiliates with access controlled by and assigned to other UCI users.

Non-Functional/Quality Requirements

Scalability, robustness, ability to cluster, performance.

- The application and backend must be able to meet the performance demands of the entire campus.