Access Management Functional Model

Overview

This document provides a general description of the components and functions of the access management component of an institutional-scale Identity and Access Management (IAM) suite. It also suggests touch points with other subsystems in such a suite. Requirements for access management functionality and operation can be written based on the terms and concepts presented in this model.

Entities Relevant to Access Management

An entity is a "thing" of interest to the institution, distinguishable from other entities of its type. The five entities most relevant to access management are

1. Subjects: things that initiate actions in online systems, of three primary types:
    a. Persons: Human actors
    b. Agents: Software components or processes, other non-person actors such as organizations
    c. Groups: Collections of Persons, and/or Agents and/or other Groups
2. Resources: things that can be the object of actions by Subjects
3. Actions: Things that a Subject can potentially do with a Resource depending on access management policies. Typically expressible as verbs.
4. Permissions: Statements of the form "Subject S can perform Action A on Resource R". Access management policies are realized by defining permissions.
    a. Permission statements may also contain "Limits", conditions that must be met for the permission to be allowed. These limits can be of many kinds, e.g. time of day or quantitative bounds
5. Roles: Sets of permissions where the Subject slot can be filled in. "Filling in a Subject in a permission set " corresponds to the informal expression "assigning a Role to a Subject".

Note that different terminology may be used in other treatments of the subject of access management, but in one way or another, these concepts are in play. [1] The above definitions will be used broadly and hopefully consistently in CIFER materials.

The Access Management Problem Space

The foundational activity of access management administration is creating and using specific instances of the above entities in a way that supports an organization's ability to ensure that resources are accessed and acted upon in line with the relevant practices and policies.

Access management systems must support delegated administration. Forcing all access management administration to be performed by a specialized group of IT staff does not scale well. *Delegation* is the capability that allows a subject to set access management policies over resources in their sphere of authority. Delegation is in fact just a special sub-type of access management policy governing who can assign which kinds of permissions to others. Through delegation, administrative controls can be pushed out and down to the appropriate level in the organization.

In cross-institutional collaborations, any type of Subject (Person, Agent, Group) or Resource may be homed in any of the collaboration partner organizations. This collaborative federation scenario brings a number of new challenges to the access management function. Subjects may authenticate at their home institution and access resources at another. In higher education today this is most commonly achieved through SAML-based identity federations such as InCommon and SAML implementations such as Shibboleth. Delegated administration of Groups and Permissions might need to support users from organizations other than the one hosting the administration tools. Resource access Permissions and policies might need to be set by admins at a Virtual Organization and then be used to govern access requests from arbitrary Subjects on Resources that may be located anywhere in the collaboration ecosystem. At present there is no comprehensive solution to the full set of use cases in the collaboration space. CIFER access management solutions must factor in requirements from this space in their design so that eventually the CIFER-associated services and tools natively support operations in distributed, collaborative ecosystems.

Typology of Access Management Functions

Function	Abbrev.	Definition
Policy Enforcement Point	PEP	The system entity or feature that intercepts a user's access request to a resource, then calls on the PDP to evaluate the authorization decision and finally enforces the PDP's allow or deny decision on the user's original request
Policy Decision Point	PDP	Point which evaluates applicable policies for a given request and issues authorization decisions
Policy Administration Point	PAP	Point through which administrators define, manage and deploy access control policies

Policy Information Point	PIP	The system entity that acts as a source of attribute values concerning subjects, resources and environmental or contextual factors that bear on policy evaluation processes of a PDP

It is not enough to define access management entities and policies. To actually control subjects' access to resources day to day, the relevant policies must be enforced at the resource end-points. In a mature access management environment this can be achieved by placing Policy Enforcement Points (PEPs) in the flow between Subjects and Resources and by having those PEPs defer to Policy Decision Points (PDPs) to evaluate the applicable permissions and return an allow/deny decision for the PEP to enforce [2].

To date, relatively few applications and services are capable of leveraging external, modularized PEP/PDP services. In the absence of PEP/PDP support, access management is typically realized by recourse to other IAM subsystems, notably Provisioning.

For any access management scenarios one can imagine, and even when nothing like formal PEPs, PDPs, PIPs and PAPs are in play, there are still policies or practices, information objects, decision points (which may be human brains) and enforcement points, all of which may be either explicit or implicit. We will highlight examples of this in the patterns discussed below.

Access Management Deployment Patterns

Given the large variety of IT components across which access control must be enforced, a number of different access management deployment patterns need to be available to realize proper access control in real-world environments. The patterns listed below are common ones. They are arranged roughly in order of sophistication from rudimentary to advanced. In each pattern there is a characteristic mapping (or collapsing) of functions (PDP, PEP, PAP, PIP) onto components (human administrators or software processes) in the system under consideration. To the extent that multiple components are involved in a given pattern, certain access management data objects need to be created and shared between them to realize end-to-end policy application and enforcement. These points are treated more fully in each of the patterns described below.

Delivering access control information to appropriate components

Web Single-Sign-on (SSO) Authentication functionality and/or Provisioning functionality play a crucial role in nearly all access management deployments. They are the means by which the relevant access control information objects are made available to the components that need them at the time they are needed.

In Web SSO models, information relevant to access control decisions is delivered as part of the authentication or identity provider protocol or service. In SAML systems, these are delivered as attribute assertions. Typical of attribute assertion information that might play a role in access control decisions are identifiers for the authenticated user, the user's affiliations, group memberships or roles, entitlements for specific services and resources and other identity attributes about the person. Any or all of these information items may be relevant to the allow deny decision at the relying party. As might be expected from the term Web SSO, this model is most appropriate to browser-delivered services and resources. The information the relying party needs to make the allow/deny decision is made available at the time and in the context of a particular user's attempt to access or act on a particular web resource.

Provisioning approaches are complementary to Web SSO approaches in the sense that they cover scenarios in which for one reason or another not all the required user-related information can be delivered in the same process stream in which the user authenticates or attempts to access the resource. Properly orchestrated provisioning processes can push information out to consumer systems in ways that leads those systems to behave in ways consistent with the access management policies of the organization. In the degenerative case in which a consumer system equates existence of a subject "account" with that subject's authorization to access system resources, provisioning and de-provisioning of local system accounts is the only way the access management function can be realized. The common feature of provisioning approaches is that some set of information that bears on access control decisions is "pre-positioned" in the place and manner that the service or resource component knows how to handle.

Local account- / record- / identifier-based access control

This is the degenerate pattern in which mere presence implies authorization. That is, if the system has a local representation of a particular subject, that subject is considered authorized to use the system. A simple example is basic access to a Unix host. If there is a user account for someone on that host, they are implicitly authorized to use the system. Another classic example of this pattern is an .htaccess file that lists identifiers/user ids of individual subjects (most often people) that are allowed to access a protected URL of a web-based service or resource. The PEP may be as simple as a few lines of code used to check for the existence of the local account, record or identifier.

Note that in many instances of this pattern, Authentication services are the basis for determining whether to accept as valid the active subject's claim to "own", or map to, a given account, local record or identifier. Note also that in this pattern human administrators must make sure the appropriate accounts, records or identifiers are in place. The only access management function that is not dependent on explicit human action is the PEP. The presence of the account or record or identifier (put there by the system administrator at the request of the resource owner, or via Provisioning), could be thought of as the pre-computed allow/deny decision that the PEP enforces. That account, record or identifier is the only access control information object in play in this pattern.

Group-based access control

In instances of this pattern, the Subject and/or Resource in an access control Permission is not an individual named actor or object but the identifier of a Group of such entities. The policy is meant to apply to all the Subjects and/or Resources that are members of specified groups. A noteworthy feature of this pattern is that it simplifies policy administration in the sense that a single Permission statement, "Group G may perform Action A on Resource R" may be applicable to an arbitrarily large number of individual Subjects and/or individual Resources. Role-based access control (RBAC) may be thought of as a specialized sub-type of this pattern. As in the previous pattern, Authentication likely plays the role of determining the identity (including group memberships) of the user/actor in the scenario.

The responsibility for creating the groups in question and managing their memberships might fall to one party and the access control enforcement to another. Human-to-human communication or Provisioning might be the way in which a resource owner or service provider learns that members of a specific group should be given such and such rights to the resource. In this case, as in the previous case, the only automated component is the PEP. The access management information object in play here is an assertion or data item conveying the group memberships for the Subject and/or Resource in play.

Entitlement-based access control

This pattern can be thought of as one in which the user/Subject's request for access is accompanied by an information object that asserts or is meant to convey that "The bearer may perform Action A on Resource R". Thus entitlements can be understood as a special sub-type of Permission.The party that creates an entitlement-style permission can be thought of as creating a canned access policy decision. Web SSO Authentication or Provisioning can then be leveraged to deliver the entitlement to the resource or service component. This pattern requires the Resource Owner/Service Provider to trust that the received/provisioned entitlement was granted in conformance with appropriate access policies. For this reason, some federated service providers may be reluctant to work within an entitlement-based access control model. In a way the federated version of this pattern takes the policy evaluation out of the hands (and eyes) of the Service Provider and leaves it in the hands of the Identity Provider. In this case, policy enforcement at the service endpoint can be reduced to a simple matter of the explicit or implicit mapping of an asserted entitlement to an Action/Resource pair.

Policy-based access control

In this model, the externalized access control functions (PEP, PDP, PAP, PIP) all play their explicit, formal roles in the access management process. This allows for the greatest degree of abstraction, centralization and/or delegation of access management functions, but it requires the deployment of all the appropriate P*P packages and services and their integration with each resource and service end-point in question.

References

- [1] Access Management Recipe from MACE-Paccman
- [2] Draft NIST 800-162: Guide to Attribute-Based Access Control