

**METHOD AND SYSTEM FOR THE
SPECIFICATION AND ENFORCEMENT OF
ARBITRARY ATTRIBUTE-BASED ACCESS
CONTROL POLICIES**

RELATED APPLICATION

[0001] This application claims priority to U.S. Provisional Application Ser. No. 61/026,743, which was filed Feb. 7, 2008.

BACKGROUND OF THE INVENTION

[0002] Access control mechanisms are a major component of any operating system and many applications. Access control policies come in numerous forms, with various methods for authenticating users, access control data constructs for specifying and managing policy, functions for making access control decisions and enforcement of policies, and a scope of protection that includes a defined set of users and resources.

[0003] One drawback of having multiple heterogeneous access control mechanisms is a lack of interoperability. Access control policies are often global and span many systems and applications. Users with vastly different attributes and credentials have a need to access resources protected under different mechanisms, and resources that are protected under different mechanisms differ vastly in their sensitivity, and therefore accessibility. This lack of interoperability introduces significant privilege and identity management challenges.

[0004] Another drawback to the existing approach to access control pertains to policy enforcement. Operating systems limit enforcement to instances of Discretionary Access Control (DAC), simple variations of Role-based Access Control (RBAC) policies, and Multi-level Security (MLS) policies. However, issues exist even within the enforcement of this narrow set of policies. DAC and RBAC are considered weak in that that users (through overt actions and mistakes) and malicious code embedded within applications can potentially leak sensitive data to unauthorized users. Also, objects are also often under-protected under DAC and RBAC alone. For example, although access to medical records may be restricted to users in the role "Doctor," not all doctors may have access to all medical records. Depending on the institution, other policies may come into play. Medical records can be classified, only accessible to those doctors in a particular ward, or accessible only under the discretionary permission of a primary physician. Additionally, MLS mechanisms can impose user and administrative inconveniences. As traditionally implemented, MLS policies impose restrictions uniformly on users and their processes thereby a user within a session is prevented access to information for which that user is otherwise legitimately authorized.

[0005] One partial solution to meet policy needs not provided by operating systems is to implement access control mechanisms within applications, such as database management systems, enterprise calendars, and time and attendance calendars. Typically, any application that requires user authentication usually includes access control mechanisms. This proliferation of access control mechanisms further aggravates identity and privilege management problems and can undermine policy enforcement objectives. For instance, although an operating system may narrowly restrict user access to a specific file, a user with read access to the file can copy the file to a message and mail the message to anyone in the organization.

[0006] Another partial solution to meet general policy needs is an OASIS' standard eXtensible Access Control

Markup Language (XACML) that describes both a policy language and an access control decision request/response language (both encoded in XML). The policy language describes general access control requirements. The request/response language allows for queries to ask whether a given action should be allowed and interpret the result. One drawback of XACML is that it does not specify or enforce policies that pertain to processes in isolation to their users, thereby disallowing the specification and enforcement of a wide variety of related policies. Another drawback of XACML is that its Policy Decision Point is stateless, which further place limitations on the policies that can be specified and enforced.

[0007] Another partial solution would be to use various configurations of Role-Based Access Control relations to simulate Mandatory Access Control and Discretionary Access Control policies. This was demonstrated by Sylvia Osborn, Ravi Sandhu and Qamar Munawer, in "Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies," ACM Transactions on Information and Systems Security (TISSEC), Volume 3, Number 2, February 2000, using the RBAC96 model. One drawback to this approach is that Osborn et al. applied a series of obligation relations in the configuration of these policies that can only exist in theory, and are not specified in the RBAC96 model. Another drawback is that their strategy for simulating DAC requires the creation of a multitude of roles that would exceed the number of objects in the system. Simulating MAC also requires the creation of role-permission assignment relations that exceed the number of objects.

[0008] Another partial solution was proposed by David Ferraiolo, Serban Gavrilu, Vincent Hu, Richard Kuhn in "Composing and combining policies under the Policy Machine," SACMAT '05, Jun. 1-3, 2005, Stockholm, Sweden. A drawback of the Ferraiolo et al. is the limitation and inefficiency in specifying and enforcing policy. The framework required the costly computation and activation of a set of user attributes for a set of processes running in a session, in order to gain access to a resource. Further drawbacks include the lack of control at the individual process level, the lack of constraints on users and processes, and the inability to dynamically alter the policy state of the machine in support of the specification and enforcement of policy.

SUMMARY OF THE INVENTION

[0009] An exemplary general attribute-based access control system includes at least one resource server, at least one client module, an access control database including basic data sets and basic relations between the basic data sets, at least one server module including an access decision sub-module that computes a decision whether to grant or deny access to computer-accessible resources referenced by objects, an event processing sub-module that processes events, and an administrative sub-module that creates, deletes, and modifies elements of the basic data sets and the basic relations.

[0010] An exemplary general attribute-based access control method includes selecting an attribute-based access control policy for specification and enforcement, establishing a configuration of basic data sets in an access control database for the selected attribute-based access policy, and establishing a configuration of basic relations between the basic data sets to control how the policy is enforced.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The various features and advantages of the disclosed examples will become apparent to those skilled in the art from