What is Access Management?

Software Guards for Integrating Applications

January 2004
www.2ab.com

Introduction

Access management is a simple concept. Every business has information that needs to be protected from unauthorized disclosure. To protect information, companies define policies that govern who can access specific classes of business and/or personal information. For example, if a manager seeks to access the salary of a subordinate, they should have authorization to do so, however, they should not be authorized to access the same information about a chief executive. That is, there is a policy that specifically governs the release of an employee's salary. Or is there? The answer is: "Probably not." What exists is a written policy related to disclosure of proprietary business information (and perhaps even a separate policy related to disclosure of employee personal information). Because human beings are skilled at generalizations, we expect someone in authority to be able to classify the request for salary information and make a decision.

Access Management software has a simple goal. It allows the human who previously acted as a guardian of sensitive information to be removed from the process without loss of access control. This sounds simple, but most businesses are struggling with the implementation of access management as they integrate and extend their applications. This is because machines cannot classify information or make access decisions unless they are explicitly programmed with algorithms to accomplish this. When you take the responsibility for access decisions away from human beings, it becomes necessary to insert software guards into your applications.

A costly problem lurks: The access policy used by software guards is often coded directly into the business application (typically requiring new database tables and/or directory infrastructure). When access policy or audit requirements change, application software must be modified, tested and redeployed. Additionally, when access policy needs to be examined or applications audited for conformance a code review is required.

A service-oriented solution emerges: Access Management solutions, as defined by this paper, provide an alternative to the costly embedding of access policy. They allow application software guards to leverage services that enable access policy to be modified, tested and deployed dynamically without application code changes. This enables your developers to concentrate on providing business software. Access management solutions efficiently enable high performance access controls in distributed environments while allowing centralized management of access policy. An Access Management solution includes programming interfaces (APIs), policy management tools and auditing capabilities.

As part of your quest for an access management strategy, consider the following questions:

1. Who should be responsible for access policy?
2. What kind of access policy do you require?
3. What resources do you need to protect?
4. How do I plug in the access management solution?

This paper describes a systematic approach to managing the complexity associated with software access management. It outlines a service-oriented architecture that maintains a clean separation of concerns between application domain functionality and access management. We hope this paper will help you define what access management means to your organization.

Why has Access Management become an Application Issue?

Often, individuals are granted access to business applications using operating system, database and/or network "access control" mechanisms. That is, the application has no responsibility for access management; application access is controlled by the runtime infrastructure. Increasingly, however, existing applications are being integrated and/or extended to an expanded base of end-users by leveraging technologies that bypass or tunnel through operating system and network security. These modernized applications not only steward business and personal information, they also span technology boundaries (e.g. the Web, J2EE, JMS, CORBA, RDBMS). It may be impossible for existing security infrastructure (not designed for multi-tier access) to maintain and communicate the identity of the user through each technical tier, making it impossible to leverage existing identity-based access control mechanisms. In fact, emerging identity management standards only address sharing identity in the e-business (i.e. "Web" technology) domain. Integrated business applications, however, are increasingly being held responsible for user and access management in service-oriented architectures that span technology boundaries to deliver functionality.

Unfortunately, when an application development group goes to their security organization for assistance (or advice) regarding the protection of business information or application features, they will likely be told: "Our security infrastructure will not help you with these issues – those are application business rules." This is because the focus of security infrastructure (and associated security organizations) has been on protecting networks and operating systems, not applications. This is understandable. Overwhelmed with attacks on their networks, corporate security groups have no resources available to assist with deployment of a security infrastructure for application-level security (often characterized as fine-grain access control).

Application security, therefore, must address any security-related requirements not provided by the runtime security infrastructure. In the areas of access management, any requirement to restrict the a) usage of application features or b) access to business and personal information, is part of "application security."

Who should be responsible for access policy?

To implement an application access management solution, you must ensure that access policies exist and are unambiguous. Although access controls will be enforced by technology, defining access policy is the responsibility of the business. For this reason, access policy related to release of sensitive information and/or application features should be documented using business terminology. During the analysis of these business requirements, concise rules will be defined governing who has access to specific classes of business or personal information and under what circumstances (there may also be rules regarding who can access application features). This analysis often requires a significant classification effort in three areas: 1) information, 2) application features and 3) people. Many companies already have an information protection group tasked with ensuring that business policies are in place to ensure the protection of business and personal information. Such an organization can play an important role in ensuring that access policy is consistent across business applications. If each application group does this classification independently, inconsistencies in policy may occur.

But can an internal organization define access policy? Increasingly the answer is no. Legislation regarding confidentiality and privacy requires that individuals be allowed to define who (and under what circumstances) personal information is released. This adds new requirements for business applications in the area of access management. The users have become policy administrators with respect to access to personal information. While the application may restrict the policy choices, it must be able to dynamically change the policy in use.

It's obvious that this issue exists in healthcare, but other domains are seeing this trend toward individuals as access policy administrators. For example, in telecommunications, newer cell phones have a GPS embedded. This means that it is possible to very concisely locate the cell phone. This is a useful and desired feature in an emergency situation. It's easy to imagine parents wanting to be able to track their children using this feature, but what if a stalker has access to this information? It's clear that access management must be part of any application that supports location-based telecommunications products.

What kind of access policy do you need?

Access Policy can be very simple or very sophisticated. Once it has been determined that applications require access management features, they typically begin with very simple access control policy based on user identity. There are many applications, however, that require sophisticated access policy. To determine your requirements for access management solutions, you should determine the type of access policy that you require.

Access Policy can be classified as follows:

Policy Type	**Question answered with regard to protected resource (information or application feature)**	**Example(s)**
Identity-Based	Are you an individual that has been specifically granted access?	User ID / Password, Private Key, Electronic Token, Biometrics
Role-Based	Are you currently in a role that has been specifically granted access?	Manager, Emergency Room Personnel
Group-Based	Are you part of a group that has been specifically granted access?	Accounting, Engineering
Context-Based	Is the context of the request such that access should be granted to this individual?	Time of Day, Location, Emergency, Account Balance
Entitlement-Based	Is this individual entitled to access this class of information?	Clearance Level
Relationship-Based	Is this individual entitled to access the personal/business information because of a relationship with the person or business?	Primary Care Physician, Manager of Employee, Account Representative, Parent
Rule-Based	Does the policy governing access to the resource allow this individual to access the resource?	Combination(s) of above

Access Management solutions may also support different types of rules. For example, iLock Security Services supports all of the Policy Types shown above and allows access policy to have multiple "rules." These rules determine whether or not to allow access. Rules are of the following types, and in a "rule-based" policy are evaluated in the following precedence order:

Rule Type	**How the rule is evaluated**	**Example of usage**
Nobody	Deny access to every one.	In a Context-Based Policy, access may be denied during certain times of the day.
Deny	Deny access to anyone that has any of these credentials (access ID, group, role).	A security alert is in place. You may wish to temporarily deny certain groups who normally have access.
Required	Allow access only if the requestor has all the credentials.	Allow only owners who are officers (you must be both an officer and an owner).
Any	Allow access to anyone with any of these credentials.	You wish to allow users who are in the group *administrators* -or- have the ID *mike* -or- are in the role *accountant*.
Anybody	Allow access to anyone.	You may wish to audit the request for the resource even though you do not restrict access.

What do you need to protect?

Traditionally, machines and networks have been the resources we protect. However, as we integrate our applications and expand the use of systems, we have seen that the application assumes the responsibilities for guarding access to business information and/or application functionality. The security community uses the generic term *resource* when discussing business information or concepts that need to be protected. Protected resources are typically given a unique name (or ID) that is used in communicating with an access manager to request an access decision. Deciding what resources should be protected and assigning them an ID sounds simple – and sometimes it is – but it can also become a time-consuming identification and data classification project not considered in the original application estimates.

For example, most companies deploying a human resources application would agree that an employee's salary should be protected. So far, so good. Now, let's assume that from a technical perspective, salary is a field in an employee record that resides in a database that exists on a server accessible via an application that supports remote client access via a network. Where do I insert the software guard, and what is the actual resource it must protect to ensure salary is not accessed improperly?

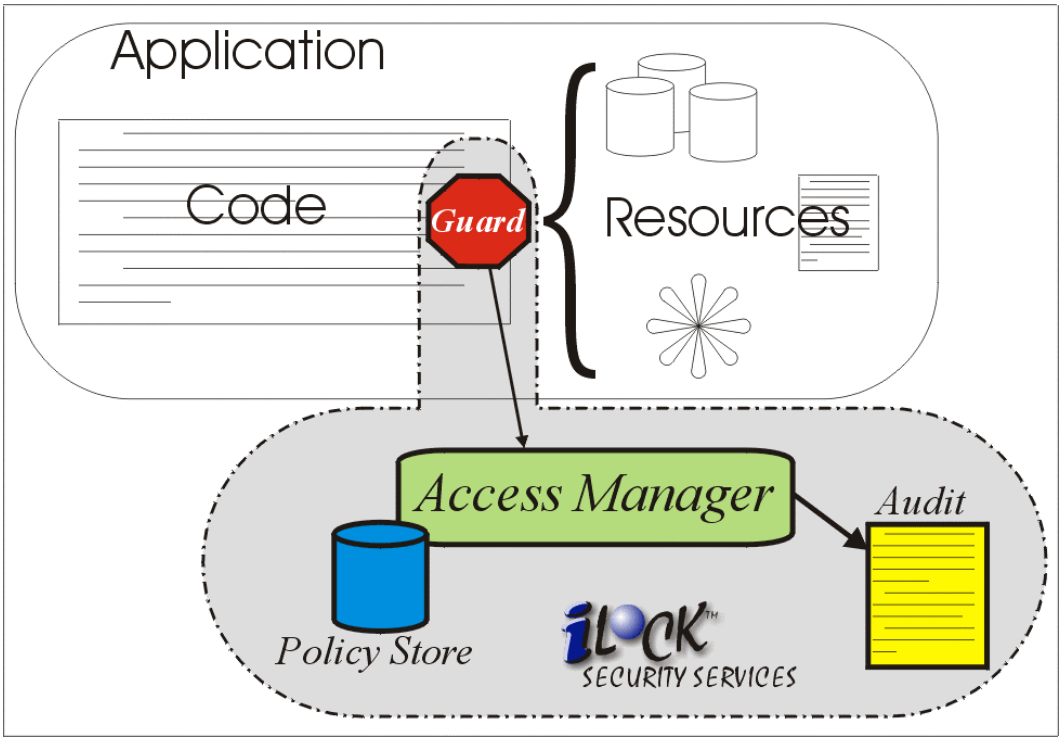
Granularity of Protected Resource	**Access Policy that protects salary**
Machine and/or network	Only people with the authority to run the HR application have User IDs on the machines where the HR application is installed.
Entire Application	Only people with the authority to view HR information are granted User IDs for the human resources application.
Specific Application Feature (e.g. Screen, Menu, Button, or URL…)	Only people with the authority to view HR information will be allowed to request salary information from the HR application.
Entire Database	Only people with the authority to view HR information have User IDs in the human resources database. The database is accessed using requestors' ID.
Table (in a database)	Only managers can view employee records.
Row (in a table in a database)	Only the employee and people in the chain of management for an employee have the authority to view an employee's record.
Field (in a Row in a table in a database)	Only the employee and people in the chain of management for an employee have the authority to view an employee's salary.
Concept (information that contains multiple fields – potentially from different sources)	Only managers have access to employee's compensation information (compensation information is a classification or concept that includes salary, commission and bonus).

How do I plug-in Access Management?

As application providers are required to implement increasingly sophisticated and dynamic access management solutions, they have begun to re-architect and isolate these features into access control sub-systems. This is a significant effort, and enterprises are discovering that application developers (with little or no security expertise) are spending an increasing amount of time building application-specific access control sub-systems instead of developing application features in their domain of expertise.

The trend toward a service-oriented architectural approach to dealing with application-level security is evident in recent analyst reports. For example:

> *META Group predicted in late 2003: "as businesses begin to put more focus on design for application securability and service oriented architecture, application-specific security mechanisms will migrate to infrastructure."*



Access Management Integration

Utilizing a standards-based access management product such as iLock, your business developers simply insert guards at the points in the software where resources are exposed. The guard consults with the Access Manager who evaluates the policy and advises the guard on allowing access.

The service-oriented model and architecture of the iLock solution was designed and standardized by security professionals for the stringent requirements of healthcare[1] access for privacy and confidentiality. Other domains, such as finance and telecommunications, have found the model also meets their requirements. iLock Security Services products provide application developer tools for Web, J2EE/Java, CORBA and C/C++ environments. It includes standard pre-built guards for protection of Web, J2EE and CORBA resources.

Can you afford to keep access policy embedded in your business applications?

[1] *Resource Access Decision Facility*. OMG Document #: formal/2001-04-01.

Challenge 2AB!

Are you still not sure if an access management solution can help? Challenge us to prove it. Send us four or five examples of your access management requirements. We'll configure an iLock Security Center with policies you can use and send you an evaluation copy of iLock complete with a working demo so you can see how to leverage iLock within your application. We'll even send you the source code for the demo so your development staff can take a look at exactly how little we had to do to insert a guard! Go ahead… challenge us. What have you got to lose – an increasingly difficult access management problem?

2AB, Inc.
1700 Highway 31
Calera, Alabama 35040

877.334.9572 (toll-free)

challenge@2ab.com