Preventative Directions For Insider Threat Mitigation Via Access Control

Sara Sinclair and Sean W. Smith

Department of Computer Science, Dartmouth College

Abstract Much research on mitigating threat posed by insiders focuses on *detection*. In this chapter, we consider the *prevention* of attacks using access control While recent work and development in this space are promising, our studies of technologists in financial, health care, and other enterprise environments reveal a disconnect between what "real world" practitioners desire and what the research and vendor communities can offer. Basing our arguments on this ethnographic research (which targets both technology and the human business systems that drive and constrain it), we present the theoretical underpinnings of modern access control, discuss requirements of successful solutions for corporate environments today, and offer a survey of current technology that addresses these requirements. The paper concludes by exploring areas of future development in access control that offer particular promise in the struggle to prevent insider attack.

1 Introduction

Threat mitigation can be reactionary or preventative When it comes to insider attack, much current work falls in the former camp: how can we *detect* it? In this paper, we pursue the latter angle: how do we *prevent* insider attack? Other chapters in this book address prevention by targeting insiders' incentives and motivation. In this chapter, we target insiders' opportunity and technical capability to execute attacks.

In particular, we focus on the electronic environment in which insiders operate. Each employee of an enterprise needs access to certain internal electronic resources (databases, file servers, programs, etc.) in order to perform her job within the context of the organization. Computer security researchers often approach the problem of insider threat assuming that an organization implements a correct access control policy; this policy simultaneously grants the user sufficient *privileges* to perform necessary tasks, yet also appropriately *constrains* her access according to the principle of least privilege (and other primitives, as discussed in Section 3). This notion implies several other assumptions about the nature of policies and the human systems they are supposed to govern:

1. For any given organization, there exists an access control policy that simultaneously grants and constrains access in a manner that is correct for that organization's goals.
2. At one point, the organization correctly identified and implemented one such policy.
3. The correctness of the policy and its implementation are maintained over time, even as the resources, users, and organization's goals change.

However, we have heard over and over---from both information security professionals and end users in industries at risk of insider attack---that these assumptions do not hold in practice. According to these reports from the trenches, currently available techniques and technology do not seem to achieve the ideals promised by access control principles and theories.

For example, organizations have shared difficulties identifying correct policies (or even determining whether they exist), as these two examples demonstrate:

- The first phase of many authorization deployment schemes requires an initial identification period in which technologists, principal managers, and users are gathered to chart out all required access and constraints. A senior information security colleague in a highly relevant enterprise regards this approach as ludicrous—business users don't have the time, and even if they did, he regards it as impossible for such a group to codify all the nuances of the enterprise's real-world operations.
- One financial services colleague laments the "access control hygiene" problem [Donner, 2001]. The need to quickly grant access leads to shared passwords and "spaghetti" access controls. No one has any idea who has access to what, or why, yet off-the-shelf access control solutions dot appear to offer sufficient agility to replace current ad-hoc mechanisms.

Enterprise partners also describe scenarios in which implemented policies do not align with enterprise goals; in these cases, users are forced to violate the policy in order to meet their job requirements; the following is a sample of the anecdotes we have documented:

- A colleague in the oil industry discussed how the security rules required a password to enter the refinery control room. However, that password is written clearly on the control room door, because practicality requires that anyone be allowed to enter; in case of a fire emergency, someone has to turn things off.
- A practitioner in the medical industry talks about having to cut-and-paste medical images from the approved image application into PowerPoint (a violation of policy), then emailing the document to an external colleague in order to receive a second a opinion in difficult cases. The policy prohibits moving images in this way because it shifts data outside the system's ability to monitor its movements, yet

the practitioner has no other way to efficiently receive the information she needs.

- An information security manager in the finance industry now insists approved data applications remain flexible and attractive---because otherwise his users move the data into convenient third-party spreadsheets and Web-based tools. When a policy interferes with getting their job done, the users move the data beyond the reach of that policy.
- A colleague reported that the Chief Information Security Officer (CISO) of a large US corporations spent the first part of each day figuring out how to “work around” new security policies---which his own group had put in place---in order to get his job done.
- A doctor serving on his enterprise’s IT committee, when hearing we worked in computer security, challenged us: was our goal to build better “IT security police,” or to help improve the lives of patients? It was clear that he was not interested in helping us achieve the former end---and that his previous experience with computer security made him suspect it had nothing to do with the latter.

(Understandably, gathering attributable anecdotes in this space---let alone solid data---is tricky, as admitting to breaking IT policy can have repercussions for both individuals and organizations.)

We have also encountered enterprises that have essentially given up on *a priori* access control altogether, as in the following examples; in these cases, implemented policies fail to provide desired constraints, but at least meet minimal privileging requirements:

- Multiple medical institutions’ policies allow every authenticated clinician to see any patient’s data; in their experience, limiting access to “need to know” is too complex, and erring on the side of excessive restriction can directly result in loss of patients’ lives.
- A professional in the power grid talks about how any person in the control room can do anything---because in the case of emergency, it would take too long to carry out the authentication process (or scramble to gain sufficient authorization if the party in question didn’t already have it).

These stories indicate that real-world enterprises have a hard time not only identifying and implementing correct access control policies, but also determining if such policies are even practically possible for their organizations. When combating insider threat, if we assume that all enterprises have correct, effective policies already, we ignore an area of research that many practicing professionals are eager to see pursued. With the belief that work in this space will improve organizations’ ability to prevent insider attack, it is this mismatch between the theory and practice of access control that we target here.

The next section of this paper identifies the types of insiders and attacks against which new research in preventative mechanisms could be useful. Bearing this threat model in mind, Section 3 provides an overview of principles and primitives on which modern access control technology is formed. Section 4 draws on our collaboration with technologists and policymakers from the financial, healthcare, and other industries to characterize requirements that drive and constrain solutions in these environments. We survey in Section 5 current access control technologies, and evaluate those solutions with respect to the requirements. Finally, Section 6 synthesizes from the survey a number of important insufficiencies of current technology, and offers ideas on how new systems or business practices could improve the state of the art. Throughout the paper we continue to share anecdotes and observations gleaned from professionals in a variety of industries; these stories help us understand better how to design our research solutions so they translate well into the real world.

2 Definitions and Threat Model

Choices of words and models allow us to highlight different aspects of a problem. Here we define a number of terms to help us narrow in on the parts of the insider problem we are targeting in this paper. We also identify specifically what types of threats we aim to mitigate.

2.1 The Insider

We define an *insider* of an organization as any person who has some legitimate privileged access to internal digital resources, i.e., anyone who is allowed to see or change the organization's computer settings, data, or programs in a way that arbitrary members of the public may not. This includes full-time employees, but may also include temporary workers, volunteers, and contractors, depending on the nature of the business. In some cases an insider may also be the child or spouse of an employee; one medical institution reported to us serious concern about doctors' families accessing medical systems through company laptops.

Note that in many cases the permission an individual has to access internal resources is not the *explicit* permission afforded to him by the organization, but the *effective* permission: a hospital may have an official rule stating that doctors may not share their laptops with their children, but also have a de facto rule of looking the other way. Organizations can implement penalties, incentives, and technology to enforce official rules and limit the set of insiders, but must also be pragmatic in accounting for effective insiders outside the formally approved set.

2.2 Types of Insiders

For our analysis, we divide the insiders who perpetrate “insider attacks” into three broad categories:

1. Those intent on malicious action,
2. Those willing to act for their personal benefit over that of the organization when the opportunity presents itself, and
3. Those insiders who inadvertently use their privileged access to do harm.

As in many areas of security, protecting against a determined and resourceful individual in class (1) is very hard. We are not asserting that better access control systems can prevent all attacks by those insiders in this category. We are, however, asserting that access control systems may make it harder for these people to do wrong, as well as reduce the opportunity and probability of harmful action by insiders in the other two categories---and that reports from the trenches support this view.

2.3 Damage of Insider Attacks

The damage to an organization by insider attacks against its electronic resources may take one or more of the following forms.

- The attacks may be destructive to systems or their availability, such as data corruption or denial-of-service attacks.
- The organization may suffer financially from the attack (either in direct costs or in lost productivity).
- They may consist of actions prohibited by law, such as insider trading or disclosure of patient health information, and thus result in regulatory fines or punishments.
- The attacks may also violate corporate rules or less formal customs, such as cultural expectations of privacy (for example, a bank employee who monitors his ex-girlfriend’s account balances, or hospital employees who read the medical records of celebrity patients).

In addition to the risk of lost business associated with destructive attacks, the cost of repair or theft, and the penalties incurred by legal violations, enterprises increasingly worry about the *reputation risk* that publicized insider attacks pose. One senior security professional of a large investment bank described to us the horror he feels at the thought of his firm receiving negative press in a major newspaper; any breach---even just a perceived breach---can dramatically impact stock prices and market shares. These complex costs associated with reputation risk are an important incentive for effective preventative measures.

2.4 Threat Model

Given these definitions of insiders and insider attacks, we will now discuss the type of scenario we will focus on for the rest of the paper; specifically, we address the concern of improper privileging.

We say that an access control policy P is *correct* for an organization O if P provides users access to and constrains users' access of electronic resources according to O's goals. These goals include business objectives, corporate policies, and regulatory requirements. A policy P that is correct for O is also *practically correct* when adopted by O if it meets the following key requirements, as discussed earlier: first, the correct policy must be logically possible (not self-contradictory or otherwise unrealistic); second, the policy must have been correctly implemented in the organization at one time; third, that correctness must have been maintained from the time of implementation to the present, across various changes in the organization and its goal; and fourth, this policy must actually match what happens in practice within the organization.

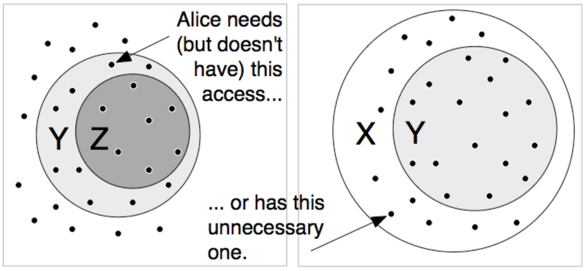
Alice needs (but doesn't have) this access...
Y
Z
X
Y
... or has this unnecessary one.

Fig. 1. Under-privileged (left) and over-privileged (right) access. The correct privileges for Alice's job are represented by set Y, and the actual privileges by Z or X.

Policy implementations that do not conform to these requirements will result in users having improper access privileges. For example, assume that the correct policy P_c provides Alice with the privileges of set Y in Figure 1. If P_c is not implemented correctly, Alice may only receive the set Z; in this case, Alice is under-privileged, and may not be able to complete the tasks her job requires. Similarly, if P_c was correctly implemented but did not adapt when Alice changed departments and started her current job, she may have access to set X, which constitutes over-privileging. Under-privileging can lead Alice and her coworkers to take matters into their own hands (with shared passwords, copy-and-pasting, and the like) in order to get their jobs done---thus leading to a *de facto* policy that not only allows over-privileging, but also moves the policy outside the realm of what's manageable.

From the anecdotes related in Section 1, and from many other conversations with practicing professionals about deploying and managing access control systems, we believe that over-privileging is a common occurrence in today's enterprise environments. Furthermore, we believe that over-privileging represents a significant source of insider threat for these organizations. The rest of this paper works to understand this issue of improper privileging, its relation to the problem of insider attack, and potential solutions.

3 Background and Primitives

The task of limiting access to electronic resources has been around almost as long as the electronic resources themselves. Scheduling algorithms in the first time-sharing systems provided multiple users with a (hopefully fair) share of CPU time, disk access, and network connectivity. In this section, we will consider the theoretical underpinnings necessary to use access control to prevent insider attack in distributed systems.

3.1 Authentication and Authorization

The computer security world defines *access control* as providing or limiting access to electronic resources (we can also say granting or limiting trust) based on some set of credentials. Access control typically consists of two components: *authentication* and *authorization*. Authentication is showing who (or what) you are; i.e., demonstrating possession of certain credentials. Authorization is the system determining if your credentials are sufficient to provide you with a requested type of access.

In many cases, we think of authentication in terms of *identity*: is that really Alice on the other side of the keyboard? In reality, there are lots of different types of credentials and properties other than identity that we can authenticate and use in making authorization decisions. For example, any valid "student" ID with Alice's photo will permit Alice to watch a Dartmouth hockey game; Bob may trust the person with the nametag at the appliance store to help him choose between features, no matter what the person's name is; however, Carlo may not trust the "valet" with the baseball cap at the hotel to take his car.

In focusing on the authentication decision---making sure that it's really Alice---it's easy to forget the subsequent mirror goal: how can we recognize when Alice is no longer there? Some real-world enterprises call this the *de-authentication* problem.

The mirror problem to authorization is *de-authorization* or *revocation*: if Bob does something bad (or simply changes jobs), how do we make sure he does not keep his now-inappropriate privileges on the system?

3.2 Access Control Principles

We quickly review the basics of access control and authorization, as they relate to an enterprise regulating how insiders access resources. (For a more thorough discussion of this material, readers should consult an introductory book, such as Smith and Marchesini, 2007.)

In the basic picture, we usually start by thinking about a matrix. Each column represents an *object:* an electronic resource. Each row represents a *subject*: an actor, such as an employee, who can take actions. Each box then lists the *privileges* that subject has to that object.

This basic model (e.g., Lampson 1974) lets us start thinking about initial principles.

The principle of *least privilege* teaches keeping each box as sparse as possible; the fewer actions one is allowed to take, the less the chance that, by accident or malice, one can cause damage. For example, the authors of this chapter often remove their own "write" permission from critical program or text files, in order to lessen the chance of accidentally modifying them while examining them with an editor.

The principle of *escalation* allows a subject to add back certain rights to an object. (Essentially, the rights-box for that object *itself* becomes an object in the matrix.) For example, when we really need to change one of those critical files, we can do so---after first adding our privilege back. Some enterprises make this process more heavyweight: e.g., by requiring the employee to explicitly request the privilege from a manager, or to explicitly acknowledge that the elevation is significant and will be audited (the latter is mechanism is sometimes called *break-glass*, used as a noun: "there was a break-glass on that record"). Some researchers have even formalized this notion as *optimistic security* [Povey, 1999].

The principle of *separation of duty* takes least privilege into another dimension: we decompose a critical action into separate pieces, and require that different subjects take these actions.

3.3 MAC, DAC, and Intermediate Schemes

Initially, we might try to use this basic matrix model to actually reason about and manage access control. Management at this extreme, raw level has its own name: *discretionary access control (DAC).* The owner of each object has full discretion on setting permissions.

However, with this approach, reasoning about or ensuring any high-level properties of what can happen in the system can quickly get out of hand. Consequently, approaches emerge to start imposing some structure and order on what happens. *Mandatory access control (MAC)* imposes strict limits on the permissions that can be granted, no matter what an object's owner likes. Usually, MAC is used in conjunction with *multilevel security (MLS),* where subjects and objects are organized into a lattice, and permissions ensure that information only flows in correct directions. (The MLS lattice was developed in the context of U.S. defense computing. Think of *clearance levels*, *compartments*, and *need to know*: an uncleared subject shouldn't read a top-secret document; a subject cleared for top-secret but only with a need to know about X shouldn't be able read a document associated with Y.) This seminal work resulted in a set of practices and principles (usually referred to as the *Orange Book*, the nickname of one of the resulting standards [DoD, 1985]) to ensure that users and data of all sorts of sensitivity levels can exist securely on the same system, even if some want to cheat---i.e., even if some users want to execute an early form of insider attack.

Many other formal models of security have been proposed, in order to bring some order to the chaos. The *Chinese Wall* is one that is particularly relevant to data-oriented enterprises. Here, a subject may have access to any object in some set---but once the subject exercises that right for one of these objects, she loses access to all the others. If Alice is standing on the Great Wall of China, she has the ability to jump to either side; however, once she jumps to one side, she can't jump over the wall to the other side. If Alice is a broker in an investment bank, she might have the ability to look at the records for client X or client Y; however, once she looks at X, then looking at Y might be a conflict of interest (depending on their relationship).

3.4 Users and Groups

In Section 2.2 above, we introduced the basic access control matrix model. In Section 2.3 above, in order to make it more manageable, we introduced some refinements and restrictions to the access rules. However, another approach to making it more manageable is to start putting more structure on the left-hand side of the matrix: how "users" map to rows.

One basic approach is to label the rows with a new construct, *domains*, and then think about users map to domains. E.g., in a UNIX-style OS, typing "sudo" to elevate privilege would correspond to changing domains to "superuser."

Another is to organize a set of users into a *group* and then use group membership to decide permissions. Traditional UNIX file permissions operate this way, although this group approach raises some annoying corner cases. What if a user belongs to two different groups, and their permissions differ? What if a user's personal permissions differ from the group's? Students learning UNIX file permissions for the first time usually get rather confused about such details, and even advanced programmers have trouble. Steve Crocker, formerly of ARPA and USC, reports that he regularly challenges system administration trainees to develop a UNIX file permission policy that matches a very natural and simple scenario from business organizational structure. Each year, each student comes with with a policy; and each year, upon closer examination, none of them actually meet the desired goal.

3.5 Roles and Role Engineering

The concept of *role-based access control (RBAC)* takes the indirection even further. As with the concept of domains, the rows in the matrix are labeled with *roles*. Depending on the design of the RBAC scheme in question, users may be assigned one or multiple roles; similarly, users in the latter set may have all their roles *active* all the time, or the scheme may constrain users to one or more active roles from their assigned set. (This capability helps provide, among other things, separation of duties.) The roles themselves may be organized into a rich *hierarchy*, with inheritance and other properties.

In the field, some practitioners use the term "role-based access control" loosely, just to refer to deciding access based on a user's job rather than their name. (RBAC standards and literature [NIST; Ferrailio et al., 1992 and 2007] present a more formal vision of what it takes for an access control system to be truly "role-based."

Existing research also offers guidance on *role engineering*, the process of identifying and managing the roles used in an RBAC scheme. Two basic approaches to role engineering, top-down and bottom-up, present different strengths; the latter tends to be quick and easy to roll out, but difficult to maintain, whereas the former approach is more time-consuming at the beginning, but offers advantages during role maintenance and management. The work in the top-down space aligns largely with requirements engineering principles, and includes scenario-driven, goals-driven, and hybrid approaches. Suggested bottom-up techniques include role clustering and discovery algorithms.

Some researchers use the term *attribute-based access control* for the somewhat similar concept of deciding access on something other than one's name.

3.6 Public Key Cryptography

Springing from the intersection of computer science and mathematics, *public key cryptography* is a tool that allows someone to take an action with their secret *private key* that can be verified by anyone who knows the matching *public key*---but knowledge of the latter does not (yet) enable one to calculate the former. Public key techniques can enable authentication directly---the user cryptographically proves knowledge of her private key. Public key can also provide glue to make other aspects of authentication work---e.g., it can enable a manager Alice to digitally sign an assertion that employee Bob should have access to the records for client C. Because requires neither shared secrets nor direct knowledge of the users (e.g., Alice's signed statement about Bob doesn't require that the record server have heard about Bob beforehand), public key techniques are attractive when authentication and authorization needs to cross boundaries within or between enterprises.

4 Requirements

Armed with the principles from the previous section, we now characterize the settings in which we hope to mitigate insider threat. We focus particularly on large corporate environments; although smaller organizations and those in other domains (particularly government) face legitimate insider threats, we believe that large corporations offer particularly fascinating challenges and opportunities for new development. Furthermore, we hope that improved access control solutions will generalize to simpler environments.

In this section we consider the functional requirements of access control in large corporations, as well as characteristics of these environments that drive and constrain the solutions actually deployed, including domain-dependent factors. We move on in Section 5 to evaluate current tools and techniques in reference to the requirements characterized here.

4.1 Functionality

The way an organization implements basic access control faculties (authentication, authorization, constraints, etc.) depends on its goals; large corporate environments' requirements of their access control systems often include the following.

Distributed authentication: resources managed by different entities throughout the enterprise should be accessible via a common authentication scheme. This means that an individual user should need a small number of credentials to authenticate throughout the organization, and not an extensive personal library of identifying information.

De-authentication: the system should recognize when a user is no longer using her authenticated session, and prevent other users from using that session illicitly.

Distributed authorization: as with authentication, resources managed by different corporate units should share a common infrastructure for presenting and evaluating authorization requests. Information used in authorization decisions may flow to distributed resources from a central source, or may be gathered in a distributed manner from a variety of authorities. (As a simplified example, consider residents of a condominium development who wish to add a deck to their unit. The community bylaws might require residents to go to the central association meeting to ask permission, or may require them to also query each of their neighbors individually before performing the renovation).

Distributed privilege assignment: The duties of issuing authorization privileges should not be concentrated exclusively in the IT department, but should generally rest with business users who are qualified make issuance decisions. This capability may be distributed to the point that average end users can assign privileges to each other; this is useful in scenarios where *delegation* of one's own tasks (and thus privileges) to a peer or subordinate is desirable.

De-authorization or privilege revocation: Corporations wishing to reduce the risk of attack by recently fired employees must be rigorous in de-authorizing users in a timely manner. Privilege revocation is also essential

Expressive Constraints: Increasing regulatory requirements combined with enterprise risk reduction strategies leads many corporations to seek fine-grained (often rule-based) constraints on access to follow principles like least privilege. Constraints may be either hard-wired (Avi can't access resource X) or dependent on additional context (if Xia has already purchased so much in supplies, she cannot put in another requisition order).

Privilege Auditing: Many corporations also aim to reduce risk by regularly auditing the privilege sets of their users. Access control systems should allow qualified managers to evaluate which users have which privileges, and potentially additional information, such as when the last time a user used a certain privilege, or how many of the user's peers share it.

Post-facto Auditing: Enterprises also need to be able to audit access control transactions that have occurred in the past, both to evaluate the success of their policies and to gather information in case of a security failure.

Many professional colleagues consider these capabilities as essential to access control systems, and vendors have accordingly attempted to integrate them in current products. However, organizations' deployment of these solutions are challenged other business requirements, which must also drive technology in this space.

4.2 Usability and Cost

Cost has long been a critical consideration for corporations deploying new technology. When thinking about cost, it's important to consider not just the monetary outlay for the obtaining, installing and maintaining the technology---it's also important to consider the impact the technology has on the enterprise's business processes. Can employees figure out how to use it? Does the technology make it easier for employees to get their jobs done, or does it slow them down? *Usability* and cost are thus closely linked in access control solutions that require interaction with users, be they non-expert employees or experienced system administrators. The motivations for usable authorization solutions are similar to those that drive quality user interface design (the usual connotation of "usability"), although the usability issues in this space extend far beyond the interface.

Usability and cost find a clear connection in the notion of productivity. Systems that are easy to use and facilitate users' doing their jobs (instead of getting in users' way) enable those users to complete work at a more rapid pace. We can expect that the introduction of new systems will reduce productivity temporarily, but highly usable solutions will improve performance after an initial period of user training and adjustment. New systems whose usability does not improve with time can result in direct reduction of productivity.

Systems that contain usability barriers also present "hassle cost," which can pose more serious concerns than reduced productivity. Basic cases include quiet boycotts of new solutions, such as nurses who continue to record patient data in paper-based notebooks instead of using laptops with too-small keyboards. Having some data in notebooks can make it more difficult for the next nurse on shift to complete his job, causing tensions in the department. In contrast, louder boycotts can involve entire departments or classes of users, and often result in tech teams being forced by upper management to change or remove new solutions. (The flow of communication between technologists and users in such scenarios can be a factor when these large-scale boycotts occur; direct feedback during early testing of a new technology often allows for usability refinement and prevents wholesale revolt.) High degrees of hassle cost engender animosity against the technology group, and can impact the success of future deployments.

Our external collaborators have also related surprising stories of users' ingenuity in circumventing systems that posed too great a hassle cost. One organization deployed proximity detectors to de-authenticate users. After an initial period of push-back, it seemed as if end users had come to accept the solution; however, the statistics regarding session length betrayed that the users were not actually being de-authenticated as desired. Further inquiry revealed that the proximity detectors had no minimum distance limit, and that a user could take advantage of this by covering a detector with a Styrofoam cup. After several iterations of refinement, the organization eventually had to roll back its deployment of proximity detectors because it could not satisfy the usability requirements of its users.

In the Styrofoam cup example, the company discovered rather quickly that users were manipulating the control technology. In other cases users have gone months or years quietly circumventing solutions that meet security requirements in order to complete their jobs in a usable way. Such practices present tremendous opportunity for insider attack.

Unusable systems will be circumvented or detested, with repercussions for employee morale (and future inclination to respect policy).

4.3 Scale and Complexity

A large corporation's scale and structure may challenge technologists' ability to deploy effective access control mechanisms. In particular, the number of employees, the geographic area over which those employees are distributed, the strength of centralized management or recognizable hierarchies within the company, and the amount and speed of change the company experiences in size and structure can all impact the feasibility of deploying practically correct policies. Table 1 provides a summary of these characteristics and the range of values that they can take on in our target settings. Below are also three small case studies to illuminate these issues; these fictional scenarios are based on observations and anecdotal evidence from collaborators in a variety of fields.

Characteristic	Range of Potential Values	
Number of employees	~10,000	~100,000+
Technology support	Fully centralized	Some or all provided by individual business units
Organizational change	Stable, low turnover	Undergoing corporate merger
Management structure	Hierarchical, single project supervisor	Matrixed, multiple dynamic project assignments, no single supervisor
Location	Single campus	Hundreds of locations world wide

Table 1 . A summary of scale and complexity issues that challenge corporations in using access control to prevent insider attacks.

Corporation Alpha has about twenty thousand employees on a single campus location. These employees are divided into specialized departments, each of which maintains a high degree of management autonomy. There exists a single technology support unit at the center of the company, but individual departments often require specialized information systems to operate effectively, and sometimes bring on additional internal support people to manage them. Despite their divergent specialties, different departments are highly dependent on each other for accurate information, and thus require a high degree of interoperability among their computer systems. Interdependence combined with autonomy creates a chicken-and-the-egg problem; individual groups cannot change solutions one department at a time without impacting others, nor is it easy to convince all departments to buy into a new solution at the same time.

Corporation Beta has about a hundred thousand employees who are spread throughout the eastern United States. The company is growing rapidly; it just completed a merger with one competitor, and another acquisition is under negotiation. The employee base, the number of campuses, the types of electronic resources, and even the sectors in which the company does business are all changing from one month to the next. The company must pilot access control technology during this period of quick evolution, yet faces tremendous difficulty in planning and deploying solutions that are sure to be sufficient over the next few years.

Corporation Gamma has about one hundred thousand employees spread throughout the world. Its overall size is stable, although much of the company is organized to morph fluidly in reaction to new needs or trends. A centralized supervisory hierarchy exists for performance review purposes, but employees may be assigned to a number of different projects and functional supervisors throughout the year. (These characteristics of Corporation Gamma match the description of "matrixed" organizations used in the business management field [Burns et al., 1993 and Anderson, 1994].)

The challenges that Corporations Alpha, Beta, and Gamma face are varied, and the solutions for each will be similarly diverse. However, they offer us an intuition for the types of scalability and complexity parameters that large organizations must consider in implementing practically correct access control policies.

4.4 Domain Considerations

From high-stakes service industries (such as hospitals) to moderately-paced retail enterprises (commercial banks) or aggressive corporate settings (investment firms), it is clear that challenges specific to an organization's domain can further confound the problem of choosing and deploying access control mechanisms. For example, a user's daily activities may vary little in one, while the other requires tremendous flexibility in performing tasks whose definition and scope change continuously. The issues presented above---usability, cost, scale, and organizational complexity--are factors in access control design across domains. However, the specific requirements that these factors produce can very with the mission and culture of an organization. We clearly cannot enumerate all the types of domain-dependent drivers and constraints, but present in this section a small set of enlightening examples.

User Expectations and Communication: The expectations of the user population can dictate many qualities of an access control system; many senior professionals (like lawyers, doctors, and business executives) are less willing to adapt to using new technologies, and many computer professionals are more inclined to use their advanced knowledge to circumvent security controls when they feel those controls prevent them from doing their jobs efficiently. (Partners in a number of industries say that security experts are usually the biggest violators of corporate security policies; they feel their knowledge of the rules entitles them to make exceptions for themselves. On the other hand, usability/security researchers report that knowing how to circumvent the rules is considered a badge of honor in IT circles---it indicates the wisdom of seniority.) Conversely, investment banks tell us that they are confident that (internal) users will let them know when they find usability issues; highly driven individuals recognize the benefits technology can have on the firm's profitability, and complain loudly when a technology makes their job harder (and often offer praise when it does the reverse).

Resource Dedication: Of course, the willingness of users to provide assertive feedback when they encounter problems is only useful when an organization has the resources and experience to make use of that feedback. The financial industry spends a large percentage of its operating budget on IT resources; large investment banks often have small armies of developers and technical support people to craft and manage information systems that will provide them a critical edge against their competitors. Efficient access to data correlates directly with profit, yet regulatory requirements and threats of insider attack also require correct access control to maintain that profitability. In this setting it is relatively easy for managers and users to see the link between successful security technology and achievement of the firm's fundamental mission: making money through ingenious manipulation of information.

Enterprise Mission: The mission of health care organizations fosters a dramatically different dynamic among technologists, managers, and users than that of investment banks. Much of the pressure in recent years for *Electronic Medical Records (EMR)* has been not from medical providers, but from medical insurers (including the federal government's Medicare program, which covers nearly 40 million Americans [ADA]). Digital information sharing presents huge benefits for medical insurers, and new laws similar to those in finance place regulatory constraints on information access. However, the role of technology and information is very different in healthcare than in finance, as is the budget supporting it. (One health care organization reports that only 3.5% of its operating budget goes to IT development and support.) The reasons behind these differences are complex and beyond the scope of the paper. It is worth noting, however, that where users in finance feel that new technology gives them a competitive edge on the market, users in healthcare often feel that new technology hinders their ability to complete their professional objectives. One doctor asked, "How can security technology help me make people more healthy?" His colleagues agreed: they were more likely to commit themselves to understanding and using access control mechanisms if they could clearly see a connection with their overarching mission as medical providers. This connection (or lack thereof) of security to mission combined with differing budgets and expertise among technical staff can constrain the number of acceptable access control solutions that will be deployable in some domains.

Urgency of Access: Another difference among domains that can drive system requirements is the urgency with which users must access information. In some settings, users who are underprivileged can be delayed in placing a retail purchase order or running a budget report---but this delay has no serious business repercussions. In other situations, users who remain underprivileged can be prevented from making a big deal by a certain deadline, which in turn results in lost profit. In other domains, insufficient access privileges can have more dire consequences; for example, not having critical medical information before a surgical operation or while handling an emergency room crisis or not being able to log in to an industrial computer during a factory malfunction can result in loss of lives, destruction of property---and legal and business consequences.

These examples are drawn from a large set of complex domain-dependent issues that can interact with an organization's ability to prevent insider threat via access control. In the following section we will survey current access control implementation and management tools; where we have found insufficiencies in these solutions, we will provide examples from the real world of how those tools could not effectively prevent insider attacks for a particular organization. We note, however, that the balance of usability, cost, scalability, and flexibility required by one organization or domain is not the same as by another. Our criticisms serve to drive future research and development, and should not be taken as a rejection of current access control technologies.

5 Tools

In this section we survey technology currently available to meet the access control requirements presented in Section 4. Some of the solutions in this section have been actively deployed in production environments for a long time, some are new and relatively untested, and some are promising but exist primarily as research results. After considering the capabilities of a variety of tools, we identify in Section 6 a number of research and development challenges that can drive the state-of-the-art in access control, and improve the ability of organizations to prevent insider attack.

5.1 Passwords: Knowledge-Based Authentication

Perhaps the first thing we think about when it comes time to implementing access control in an enterprise is authenticating the users. For many users, the most obvious approach is to authenticate via something one knows---and this is typically a *password* or longer *passphrase*. (Other knowledge-based schemes exist, but passwords are perhaps the most common in the workplace.) From the point of view of developers, password-based authentication has many nice properties. It's a well-understood technology with many mature software tools (ranging from simple OS utilities to Kerberos [Neuman, 1994] and beyond); it's easy to implement; and users understand it. However, it has many downsides as well.

One of the main ones is that strong passwords can be hard for users to remember. This leads to no end of security problems. Unless forced otherwise, users will pick weak passwords that are easy to remember. Users will re-use the same password for many accounts. Users will help themselves remember stronger passwords by writing them down on post-it notes or on the back of keyboards. (An information security manager at a large firm at risk of insider attack noted that, when strolling through a workplace, would nearly always find passwords written underneath keyboards.) Users will forget passwords they don't use often; enterprises forcing users to change passwords regularly incur increased help desk costs shortly after password-change day.

An aspect of passwords that is either bad or good (depending on one's point of view) is the fact that users can easily share them other users (and often do---for chocolate [BBC, 2004], for plastic dinosaurs, squirt guns, or just because someone somewhat official asks [Smith, 2004]). From a strict security perspective, this is bad: an access control policy doesn't do much good if an enterprise can never be sure exactly "who" a particular user really is. However, from a business perspective, this can be good: when an end user needs to delegate some privilege to a colleague, she can easily do so. (When users confess password-sharing to us, it's always to achieve a reasonable business goal; it's just that breaking infosec policy was the most efficient or perhaps the only way of doing it.) Of course, using passwords for delegation has security drawbacks---users give away everything, not just the privilege in question, and do so in a way that is not easily revocable or auditable.

Providing de-authentication in password-based systems is challenging, yet vitally important in environments of mobile users and shared workstations. (One senior medical colleague reports that, in the beginning weeks of their training, new interns find that much embarrassing email---as well as requests for particularly undesirable "on-call" hours---are generated from their email accounts if they fail to de-authenticate in some departments.) One approach to de-authentication might be to require Alice to type her password in every time she wishes to take an action; however, this would quickly become unusable in the majority of corporate environments. Another mechanism that is commonly used is a timeout, whereby the user's login session is terminated after a fixed period of inactivity. IT managers in some domains lament that no single *timeout* is correct across the organization; any value they choose will present an unacceptable hassle cost to some segment of the user population. It would seem that alternative de-authentication solutions, such as those in the biometrics and tokens discussions below, are necessary in domains where timeouts are insufficient.

5.2 Biometrics: Physiology-Based Authentication

In addition to identifying users based on "something they know," another approach is to authenticate users via "something they are." (Security textbooks also teach multifactor authentication: authenticating users by using more than one of these approaches to provide extra assurance.) Common techniques here include fingerprints, hand geometry, voice recognition, and even retina and iris imaging. In theory, biometrics have usability and security advantages. It's much harder for a user to forget a thumb than a password; it's also much harder to lend one's thumb to a colleague for a while. However, there are downsides as well. The effectiveness of the biometrics always seems to be in doubt; it seems vendors tend to claim stronger reliability than reality. Users also can find them intimidating or awkward to use (e.g., Sasse 2007). Another issue in many enterprise settings is whether a user's biometric will always be available or readable. How does one do fingerprint recognition or voice recognition on a masked and gloved medical technician in an operating room?

Biometric methods of de-authentication can detect when a human body is no longer present (and thus trigger the logout of the affiliated user). For example, pressure-sensitive mats, body heat sensors, and proximity detectors might all be useful solutions. Unfortunately, the same IT managers for whom session timeouts were not successful also experienced difficulties getting commercial proximity sensors to work effectively; the sensors either lead to false negatives (logging the person who temporarily stepped out of range) or false positives (leaving departed Alice logged in because her machine faces a busy corridor). This kind of technology might be more useful in domains where users did not need to step briefly away from the computer, or where computers were not surrounded by so much traffic.

5.3 Tokens: Possession-Based Authentication

In the standard security textbook mantra, the third main approach to authenticating users is via a *token*: something they possess. Token-based authentication is common in many workplace environments: employees carry and display badges, or carry identification cards in their wallets. These tokens often can directly interact with the enterprise's IT infrastructure: for example, a badge might have a machine-readable bar code, or use *radio frequency identification (RFID)* to identify itself without physical contact. (The RFID approach raises some interesting opportunities and privacy challenges, because the enterprise can easily interact with an employee's token without the employee even being aware. This can help the enterprise find that critical manager when they need her; however, a perceived loss of privacy can also negatively impact employee morale.) Tokens can also interact over direct electronic connections; *smart cards*---credit-card-sized cards with small integrated circuits---communicate over standard electronic contacts, whereas USB devices utilize the common device interface to connect to computers. Newer technology such as *Bluetooth* can move this more involved interaction to radio.

Some tokens automatically enable de-authentication because of how they communicate. USB tokens must be plugged in to a computer for them to be used; once they are removed, the computer knows the user has finished his session. Similarly, a computer who performs authentication using RFID badges can recognize when a given badge is no longer within range. Of course, de-authentication in these examples requires the user to remember to take her token with her when she leaves; however, corporate users frequently require ID badges, keys, or other objects to do their jobs, so such a requirement seems reasonable for many environments.

5.4 PKI: Authentication via Digital Certificates

Discussion of enterprise authentication can also broach the topic of *public key infrastructure (PKI)*. In order for an enterprise to use the public key techniques of Section 3, it needs supporting glue---public key *infrastructure* is the term used for this glue. Typically, PKI begins with a *certification authority (CA)* issuing a *certificate* that can tell Bob what Alice's public key is---that is, if Bob believes such statements from this CA. A tricky aspect of PKI is *revocation*: declaring that a particular certificate should no longer be considered as valid. Many standard revocation techniques exist and are deployed; the primary approaches are, before accepting a certificate, to check a published (but perhaps outdated) list of revoked certificates, or to check with an *online certificate status protocol (OSCP)* service. In the field, many IT managers still regard it as a not-completely-solved problem. In military and industry deployments, the bandwidth necessary for unexpectedly huge *certificate revocation lists (CRLs)* almost crippled networks. As for OCSP, why bother having public key certificates if one has to check with some backend *directory* every time to see if a given certificate is still valid? Revocation---as well as the problem of key mobility---is a significant hassle.

Some enterprises use PKI explicitly: setting up keys for their users and educating them about their use. (Since humans tend not to be good at cryptography, these keys hide within other devices, such as the USB tokens discussed above, or even exist protected within the user's computer in a software *keystore*.) Other enterprises use what we term "stealth PKI": using PKI foundations to enable other authentication techniques, such as smart cards or badges, but hiding the existence of the underlying PKI from the users.

Current PKI tools operate almost exclusively on *identity certificates* (usually in the *X.509* format [Housley 2002]), which bind user identities to public keys. X.509 permits *attribute certificates* binding other properties instead; X.509 can also permit an end user to create a special *proxy certificate*. However, common Internet tools do not support these alternatives gracefully; the inability of X.509 in practice to speak about things other than names, and to let end users do spontaneous delegation to each other, are significant obstacles to using PKI to solve the access control problem in enterprises.

5.5 Distributed Authentication and Identity Management

For much of the history of access control, users' identities have served double duty as both authentication and authorization credentials. As such, the community has developed extensive technology for *identity management* tasks, which include adding, changing, removing, and auditing user accounts and their associated access privileges. (Systems that decouple authorization from identity also manage additional information; we discuss distributed authorization in sections below.)

Centralized identity management allows an enterprise to streamline access control operations across distributed systems: instead of resources *A*, *B*, and *C* all having independent set of accounts, credentials, constraint policies, and administrators, they share a central database that contains up-to-date user information. This database often takes the form of a *directory*. Standards movements in the early history of the Internet yielded the X.500 directory services specification [Chadwick 1994]). The X.500 community envisioned a global "directory in the sky," which would act as an omniscient phonebook and include all the information needed to securely communicate with anyone, anywhere, including users' public key certificates.

Scalability and privacy issues prevented this global directory from becoming a reality, but the concept lives on in the form of *Lightweight Directory Access Protocol (LDAP)* directories. A number of software vendors (as well as open source software groups, such as OpenLDAP) offer LDAP products; in particular, Microsoft's Active Directory (AD) seems to have a large market share among corporate collaborators, largely because it thoroughly integrated with the Windows operating system.[1]

Solutions like AD can act as a repository for a variety of identity-based authentication systems. In theory, AD can integrate with Kerberos as well as PKI implementations. (In practice, PKI implementation in AD is still too immature to be sufficiently usable and reliable for most corporate environments.)

When deployed widely in an organization, both systems like Kerberos and those based on a PKI can offer *single sign-on (SSO)* capabilities to end users. For example, once a user has authenticated[2] to Kerberos via AD, the server issues her computer a Kerberos ticket; she can then use the ticket to authenticate to a number of resources within the company. A PKI user who keeps his credentials in a software keystore or hardware device only needs to unlock it once in order to use his private key multiple times on a single machine. Transparent SSO can boost productivity and lower hassle cost, but requires that the de-authentication problem be addressed: how do we prevent another user from coming along and using the

[1] We note also that much Active Directory functionality is outside the LDAP specification; it is billed as an LDAP-*compliant* general directory service.

[2] Of course, the same risks posed by passwords, biometrics, or smartcards apply when those credentials are used for widespread distributed authentication; credential compromise in such settings poses tremendous risk for the organization.

ticket or unlocked keystore? This is less of a concern with hardware devices, at least once users gain the habit of keeping the device on their person. Portable keystores also provide *credential mobility*, which can help improve usability in environments such as hospitals---again, as long as users succeed in keeping their credentials with them at all times. However, another aspect of this problem is beginning to receive much attention: the ability of a malicious Web site to quietly borrow credentials from a user's keystore, software or hardware.

5.6 Distributed Authorization

As noted before, traditional access control products have viewed *authorization* almost exclusively in terms of *authentication* of a user's identity by a resource owner. However, in large environments like enterprise IT the picture is often more complex. For one thing, the "resource owner" may not necessarily be in a position to decide whether the user in question should receive access; database administrators are rarely qualified to approve an investment banker's access to certain account data, although the banker's supervisor may be sufficiently informed. For another thing, scale and complexity requirements often require additional layers of abstraction beyond a user's identity, such as roles, to manage the company's compliance with regulatory access constrains. Furthermore, the maintenance that goes into keeping such an infrastructure running is often complex beyond an individual human's understanding.

Trust Management Systems

Researchers encountering issues associated with distributed authorization have proposed extensive *trust management* infrastructures, such as PolicyMaker system [Blaze et al., 1996] and its successor KeyNote [Blaze et al., 1999]. Essentially, these systems develop formal ways to express access control policies and formally evaluate whether the requester's credentials merit authorization.

Privilege Management Infrastructures (PMIs)

The PKI community has also developed tools to adapt to the modern complexities of distributed authorization. As noted earlier, generic PKIs operate on identity certificates that bind public keys to identities; a *Privilege Management Infrastructure (PMI)* instead has *attribute authorities* (instead of CAs) issue attribute certificates that bind public keys to other attributes, such as "Employee," "Student in CS 101," "The Dean's Assistant," or even "Bob says is permitted to see record X." PERMIS---an academic project that has been piloted in some European civic applications and in GRID distributed computing---is a good example [Chadwick, 2002].

Distributed Policy Decision and Enforcement

Consequently, we see architectures for distributed authorization emerge, with *policy enforcement points (PEPs)* consulting *policy decision points (PDPs)* about whether to grant a request, and with policy languages---such as the *eXtensible Access Control Markup Language (XACML)* to express these policies

Active Directory

Many commercial tools have evolved to include features to address these complexities. For example, Active Directory supports a wide variety of capabilities beyond that of a traditional directory of users. Administrators can create *Organizational Units (OUs)*, which are composed of users, security groups, computers, and other OUs. *Group Policy Objects (GPOs)* allow AD administrators to manage privileges of both users and computers assembled into logical groups. GPOs interact with the Windows operating system to enforce constraints on file system access, trust policies (for example, what PKI certification authorities to trust), and application usage. The recursive nature of OUs allow administrators to define hierarchies to facilitate privilege management throughout the enterprise; Active Directory also provides a scripting capability to streamline tedious tasks associated with user and group privileging. By specifying a user, computer, group, and/or GPO, administrators can both audit the effective privileges in place and model the effects certain modifications would have.

However, the flexibility offered by Active Directory seems difficult for enterprises to harness. Richards et al. discuss in their administrative guide [Richards et al., 2006] the business case for migrating to an AD-based system: “Will it reduce your Total Cost of Ownership (TCO)? It sure will, but only if you design it correctly. Design it the wrong way, and you'll increase costs.” The authors quantify the scalability of the OU/GPO infrastructure by noting that Microsoft recommends organizational hierarchies of depth at most 10, and by relating their personal observations of significant slowdown when more than 12 policies are applied. This latter estimate is supported by the experience of one of our partner companies, who laments that their single greatest source of hassle cost is the lag users experience between logging in and actually being able to use a workstation. (This hassle also feeds into users’ unwillingness to voluntarily de-authenticate from a computer they are likely to use again in a short time.)

Role Engineering and Management

Most modern access control solutions implement some form of RBAC (described in Section 3.5). Although support for more advanced capabilities---like role hierarchies---has been rare, many vendors are introducing new features to meet regulatory and risk-mitigation demands. With the addition of these much-desired features, however, has also come the need for new technical solutions to role engineering and management problems.

As outlined earlier, the process of defining roles for an organization can be top-down, bottom-up, or a hybrid solution of the two. The first option is traditionally process-based, light on technology but heavy on interviews and scenario building. Our collaborating practitioners as well as a number of distinguished RBAC experts [Ferrailio et al., 2007] find top-down definition to be both tedious and difficult---if not impossible---in some scenarios. Vendors such as Bridgestream (recently acquired by Oracle, http://www.bridgestream.com/) and Eurekify (http://www.eurikify.com) offer a suite of applications to facilitate continuing role management as well as initial bottom-up role engineering. Products in this space include solutions for *role mining* or *role discovery*, whereby a clustering algorithm generates a set of candidate roles from users' existing privileges. Blindly clustered sets run the risk of becoming quickly outdated as the organization changes over time, however, and vendors are refining their approaches to include additional sources of information, such as organizational hierarchies or users' job titles. One financial institution we partnered with has dedicated significant resources to studying the capabilities of products in this space, and while the institution reports that they are going in the right direction, it also laments that most solutions are too immature and unproven to be deployable in live organizations.

Federation
The above discussions implicitly assumed that, even if distributed, authorization decisions needed to be made within the scope of a single enterprise. In practice, users and resources may be distributed across multiple enterprises, which raises the issue of how to *federate* different authorization systems. In the academic space, *Shibboleth* (http://shibboleth.internet2.edu/) is well-known example, used to allow individual institutions to maintain their own legacy ways of authentication, but let their users access resource at remote institutions. The *Liberty Alliance* (http://www.projectliberty.org/) is mainly industrial consortium devoted to open standards for Federation.

6 Ongoing Challenges

Thus far, this paper has surveyed existing research and development in access control, with a particular focus on the applicability of this work in preventing insider attacks in large corporate environments. Section 2 defined the specific insider threat model against which we aim to defend; Section 3 presented background principles of access control to elucidate the theoretical capabilities of these systems. Section 4 presented the functional, cost, usability, scalability, and complexity requirements that the threat model demands, and Section 5 surveyed some current access control tools (both research projects and commercial products). Overall, both the theory behind access control and the systems that implement it seem to be well developed.

Nonetheless, even with these principles, we still have an insider threat problem; armed with these tools, our colleagues in the trenches still report an inability to have accurate IT policy in practice. The natural response is: Why? Do the tools fail to accommodate some critical aspect of the real-world requirements? Have the basic principles overlooked something? Is it just a matter of economic incentive for a vendor to bring the right technology to the right market? To use the terminology of Section 2, would a “practically correct” access control system even reduce the incidence of insider attack? Is such an access control system possible? If not, what are the limits that bind it, and how close to the ideal can we get?

6.1 A Snapshot of a Motion Picture

Accurate access control policy requires an accurate vision of what users, roles, and permissions should be. However, real-world enterprises report that the dynamic nature of the real world makes it hard to capture a clear vision.

As discussed earlier, organizations attempting to deploy role-based access control solutions can choose top-down, bottom-up, or hybrid approaches to role definition. Bottom-up clustering algorithms succeed in grouping users using existing privilege assignments and limited organizational information, but it is not yet clear whether this approach generates roles that will be useful throughout the various changes employees and enterprises undergo. Hybrid solutions integrate the strengths of the top-down approach: build on existing task- or requirement-oriented practices to define roles, which although tedious and expensive in personnel costs, results in role sets that will gracefully evolve with the organization. Unfortunately, many organizations (often those renowned for their agility and adaptability to new business climates) are founded on dynamicism and matrixed structure; any product of a top-down methodology will be out of date before the process is finished!

Technologists at companies who experience such dynamicism thus report being forced to choose between roles that are difficult to manage (and likely to become inaccurate quickly), and roles that are more likely to evolve with the firm, but start off being incorrect at their initial deployment. We thus wonder, what approach should these enterprises take to role engineering? What ways can solutions integrate top-down and bottom-up methods to generate roles that are both accurate and that will evolve with the organization? Finally, what further research and development is necessary before vendors can realize such integration in commercial products?

6.2 Privilege Issuance and Review

In addition to the roles or other structures necessary to manage its access control system, an organization must also define the ways in which users acquire privileges, and design methods to effectively audit users’ privilege sets for correctness. Some organizations report tremendous difficulty identifying which manager or administrator should be in charge of privilege issuance to a given set of users (maybe Anya is best qualified to decide whether or not a given doctor needs a certain privilege, but Sergey knows best when it comes to nurses). Defining rules for practices like delegation or temporary privilege assignment are even more challenging, yet some, like break glass, are vital to an organization's ability to meet its most fundamental goals. (Of course, additional flexibility in privilege issuance results in less supervised control over the process, which in turn can increase risk.) Access control tools often allow such features in theory, but actual attempts to distribute issuance to end users seem to often fall short of desired functionality.

Beyond the challenges of privilege issuance, colleagues report that many managers experienced unanticipated difficulty in verifying the correctness of a user's privilege set. Although Andy may be Liz's supervisor on a given project, that doesn't mean that he will be able to review a list of her privileges and be able to identify the subset necessary for her to complete her assigned project tasks. Indeed, one organization reported that users were not able to identify which of their privileges were essential to their *own* jobs. How can we improve privilege review technology to better enable these vital business practices?

6.3 Auditing and Visualization

Privilege review allows an organization to verify correctness of one aspect of their access control system. However, corporate partners lament the lack of tools to help them maintain a broader understanding of the system's operation, and of the subtle effects of different policies. Enterprise-wide access control systems, especially those that implement privilege constraints like separation of duty, can be more complex than any computer network or technical program; we do not ask administrators to verify network architecture correctness by watching traffic flows, so how can we expect them to perform similar tasks in access control? Desired functionality in this space includes high-level cost assessment (both monetary and hassle), risk evaluation, troubleshooting assistance in case of improper privileging, and capability modeling to understand the impact a given set of policy changes would have on the enterprise's ability to meet its goals. Effective solutions to these problems will likely involve significant work in interface design and usability testing.

6.4 Role Drift and Escalation

An access control policy must adapt to organizational changes to maintain correctness over time. Enterprises that deploy role-based systems must ensure that roles are properly assigned to users, but must also make sure that roles contain proper privileges as new resources become available and old ones are phased out. How can technologists identify when a role is "drifting" away from its original definition, when it is appropriate to split or merge roles, or even when a new round of role discovery and definition is warranted?

Some domains require users to have the ability to escalate their privileges in certain situations; for example, health care professionals talk extensively about "break-glass" features. How can an organization implement this escalation requirement while still limiting insider threat? How can it evaluate tradeoffs in balancing the need to get the job done (where failure can mean dire repercussions) with the risk of insider attack that uninhibited access poses?

6.5 Expressiveness and Need to Know

As noted, many researchers assume that, by definition, the appropriate policy exists. Others assume that "proper authorization" will always allow "unauthorized action," and proceed to define insider attack that way: unauthorized action by authorized individuals. Both views suggest (conflicting) assumptions that should be questioned. We have already seen that "correct" polices tend to not to exist in the field. Are they even possible? Alternatively, why is it that "unauthorized action" cannot be restricted by better authorization? There seems to be an implicit assumption that the behavior exploited by insiders cannot be expressed within a policy language. Is that true? If not, how closely can feasible IT policies approximate the "true" policy?

6.6 Incentives

It's probably true in general that experts in an area implicitly assume that the entire population shares their belief in that area's value and importance. Computer security is no exception. Reports from real users in real-world enterprises show a diversity of opinions about whether techniques such as authentication, access control, and general security hygiene help or hinder users getting their jobs done. To the extent that users perceive a lack of alignment between security technology and their core mission, security technology will not be effective.

This line of thinking suggests that research is needed into why this perceived misalignment exists. Would it be solved by better user education? Or is the technology itself fundamentally misaligned in some way?

7 Conclusions

In the previous section we considered a number of specific challenges; we believe that these issues are representative of the types of difficulties our partners in industry have had in using access control to prevent insider threat. It is not clear at this time whether current solutions can meet these challenges as they stand, whether we need new development efforts based on current principles, or whether we need new lines of research altogether. In any case, additional work in this space will almost certainly improve the ability of large enterprises to defend against the insider threat; unfortunately, as the authors of other chapters remark, it is difficult for researchers to gather actual data from real-world partners to inform this development effort. Technologists and corporate policy makers are happy to share anecdotes and general trends, but hesitant to offer attributable facts or hard information that could pose reputation risk.

Despite the difficulty in obtaining hard data, work in this space continues; in looking forward, we wonder: what degree of mitigation can we eventually hope to achieve? Can all insider threat be prevented with well-designed access control mechanisms? We conjecture to the contrary that no access control mechanism alone can protect against attacks executed by trusted individuals *using only the privileges deemed necessary to get their job done*. Other measures that offer disincentives against abusing their privileges can mitigate the threat, but there are some scenarios in which insiders must be trusted to use their better judgment in their interactions with electronic resources.

The concepts, survey, and ideas presented in this paper deals with insider attack prevention, whereas much current work in this space (indeed, most of this book) focuses on detection. We intend our focus on prevention to complement, not replace, that the detection efforts. Better prevention can simplify the problem space that detection must address; we recall the early history of the theory of safe systems [Harrison et al., 1976], where "detection" was in fact not computable until the problem space was constrained. History offers hope.

Acknowledgments

This chapter describes work resulting from a research program in the Institute for Security Technology Studies at Dartmouth College, as well as under the auspices of the Institute for Information Infrastructure Protection (I3P), which is managed by Dartmouth College. Both programs are supported by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security, the I3P, or Dartmouth College.

In addition to our grant sponsors, we would also like to thank our partners in the financial, health care, and other industries, who must unfortunately remain anonymous here. We also thank Christopher Masone for his willingness to perform a late-breaking review of this chapter.

References

[1] American Dental Association. "Insurance: Medicare and Medicaid," *ADA Official Website*. http://www.ada.org/public/manage/insurance/medicare.asp.

[2] Anderson, R. E. "Matrix Redux," *Business Horizons*, Nov.-Dec. 1994, 6-10.

[3] Blaze, M; Feigenbaum, J.; Ioannidis, J.; and Keromytis, A. "The Role of Trust Management in Distributed Systems." *Secure Internet Programming*. Springer-Verlag LNCS 1603, pp 185-210. 1999.

[4] Blaze, M.; Feigenbaum, J.; and Lacy, J. "Decentralized Trust Management." *Proceedings of the 1996 IEEE Symposium on Security and Privacy*. pp. 164-173.

[5] British Broadcasting Corporation. "Passwords Revealed by Sweet Deal." *BBC News*, UK Edition, April 20, 2004. http://news.bbc.co.uk/1/hi/technology/3639679.stm.

[6] Burns, L. R. and Wholey, D. R. "Adoption and Abandonment of Matrix Management Programs: Effects on Organizational Characteristics and Inter-organizational Networks." *Academy of Management Journal*, Vol. 36, 1, 106-139.

[7] Chadwick, D. "The PERMIS X.509 Role Based Privilege Management Infrastructure." *7th ACM Symposium on Access Control Models and Technologies (SACMAT 2002)*. 2002.

[8] Chadwick, D. 1994. *Understanding X.500: The Directory*. London: Chapman & Hall, Ltd.

[9] Chadwick, D.; Otenko, A.; and Ball, E. "Role-Based Access Control with X.509 Attribute Certificates." *IEEE Internet Computing*. March-April2003.

[10] *Department of Defense Trusted Computer System Evaluation Criteria*. DoD 5200.28-STD. December 1985.

[11] Donner, M.; Nochin, D.; Shasha, D.; and Walasek, W. "Algorithms and Experience in Increasing the Intelligibility and Hygiene of Access Control in Large Organizations." *Proceedings of the IFIP TC11/ WG11.3 Fourteenth Annual Working Conference on Database Security*. Kluwer, 2001

[12] Ferrailio, D.F. and Kuhn, D.R. "Role Based Access Control." *15th National Computer Security Conference*. 1992.

[13] Ferrailio, D.F.; Kuhn, D.R.; and Chandramouli, R. 2007. *Role-Based Access Control*. Norwood, Massachusetts: Artech House Publishers.

[14] Harrison, M.A.; Ruzzo, W.L.; and Ullmann, J.D. "Protection in Operating Systems." *Communications of the ACM*. 19(8): 461—470. 1976.

[15] Housley, R.; Polk, W.; Ford, W.; and Solo, D. 2002 *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. Internet RFC 3280.

[16] Lampson, B.W. "Protection." *ACM Operating Systems Review*. 8(1): 18—24. January 1974.

[17] NIST. *Role Based Access Control*. http://csrc.nist.gov/rbac/

[18] Neuman, B. C. and Ts'o, T. "Kerberos: An Authentication Service for Computer Networks." *IEEE Communications*,. 32(9):33-38. September 1994

[19] Povey, D. "Optimistic Security: A New Access Control Paradigm." *Proceedings of the 1999 New Security Paradigms Workshop*. 40-45.

[20] Richards, J. ; Allen, R. ; and Lowe-Norris, A. G. *Active Directory*, Third Edition. O'Reilly Media, January 2006.

[21] Sandhu, R.S.; Coyne, E.J.; Feinstein, H.L; and Youman, C.E. "Role-Based Access Control Models." *IEEE Computer*. 29(2): 38—47. 1996.

[22] Sasse, M.A. "Red-Eye Blink, Bendy Shuffle, and the Yuck Factor: A User Experience of Biometric Airport Systems." *IEEE Security and Privacy*. 5(3): 78—81. May/June 2007.

[23] Smith, S. W. "Probing End-User IT Security Practices---via Homework." *The Educause Quarterly*. 24 (4): 68—71. November 2004.

[24] Smith, S. W.; and Marchesini, J. 2008. *The Craft of System Security*. Indianapolis, Indiana: Addison Wesley Professional.

[25] Weeks, S. "Understanding Trust Management Systems." *Proceedings of the 2001 IEEE Symposium on Security and Privacy*. pp. 94-105.