Secure Systems
Editors: Patrick McDaniel, mcdaniel@cse.psu.edu
Sean W. Smith, sws@cs.dartmouth.edu

What's Wrong with Access Control in the Real World?

Sara Sinclair and Sean W. Smith
Dartmouth College

Effective security requires looking at an entire system, as this department has noted in many previous installments. Looking at only one piece leads to security trouble—and this dangerous reductionism extends to looking at only what we pretend a system does, rather than what it actually does in the real world.

The CIO of NASA, Linda Cureton, recently blogged about information security[1] and made similar observations. She noted that the IT community struggles with computer security because of "our belief that we are able to obtain this ideal state called—secure." However, to successfully secure a computer system in practice, an IT professional needs to know what's actually going on. Cureton argues that striving for perfect security gets in the way of understanding the computer system's true state.

We aren't CIOs—rather, we're a pair of information security researchers in the academic world (currently). But we've been talking to CISOs as well as ordinary users, and, for several years now, we've been studying access control via extensive fieldwork in real-world organizations, including financial services corporations and healthcare settings. These are environments in which regulation plays a major role in infosec decisions: the confidentiality, integrity, and availability of data, code, and services are important enough to larger society that government has decided to step in to try and protect it.

During this fieldwork, we've seen the same thing that the NASA CIO observed: a wide gap between what the IT security community believes and what's actually happening in the real world hampers effective security. In this article, we speculate on some possible reasons why.

The Access Control Problem

Access control is a fundamentally hard problem. Analog human systems (such as corporations, partnerships, and families) use countless mechanisms to encode and enforce standards of resource use. Most of these mechanisms are what computer scientists would consider "informal": together, they form a complex ecosystem of sociological incentives and psychological motivators that are wholly divorced from the (sometimes) provably secure computer protocols our community treasures.

At some point over the past decades, resources moved to electronic settings—and the problem of how to provide appropriate access control came under the "computer security" domain. To gain traction representing these complex policies in formal computer terms, the infosec research community approached the challenge as any good scientist does: first, we start with a simplified model. We assume that the world is less complex, convince ourselves that we can solve the problem in this simplified world, and then move on to real-world complexity.

However, somewhere along the way, our community forgot that the simplifying model wasn't the same as the real world. We decided to go ahead and implement systems based on assumptions our simplifying model made and forgot to make allowances for real-world messiness. (The academic research community, with its focus on occasionally provably secure systems and new, elaborate schemes for expressing increasingly complex access control policies, is particularly responsible for this focus on the theoretically "secure.")

When physicists' empirical results deviate from those their theoretical models predict, they don't simply ignore the deviation; they mine it for information because they know it can help them validate and refine the model. In particular, they don't castigate the universe for being uncooperative or not sufficiently well-trained!

However, when our implemented computer security policies go awry, we often don't seek to understand why. First, we cope with the resulting crisis; next, we argue to interested parties that our system did, in fact, follow all accepted best practices and that the failure was due to some uncontrollable, external factor, such as misbehaving users.

Organization-Wide Doublethink

The result of this collective habit is nothing short of organizational doublethink. Everyone on the inside knows that the systems aren't working, but admitting this knowledge in and of itself opens them to liability—the system's perceived success depends on ignoring the problems.

Professionals from multiple large financial institutions have acknowledged this doublethink off the record; one (whom we'll refer to as Alice) related a particularly telling anecdote. While undergoing an audit by federal regulatory authorities, Alice's team worked to demonstrate that the company met the government's data security requirements; the auditors asked the expected questions and ticked the expected boxes on their checklists. Alice was elated that the reviewers seemed satisfied with her team's answers—until it dawned on her that those auditors were asking the same easy questions when they evaluated the security of other financial companies, including the banks to whom she trusted her own personal finances. At that moment, she said, she wanted to grab the reviewers and tell them what questions they should be asking, to say "Hey, aren't you curious about this information over here?" Of course, because she knew the organization that employed her didn't have satisfying answers to some of these more revealing questions, she restrained her urge.

Clearly, a security system's actual success depends on thinking honestly about it; in this situation, we must first admit that the doublethink is doublethink. As a step in that direction, this article tries to enumerate some of the simplifying assumptions the security community has made in its effort to gain traction with the access control problem. For many environments, a dramatic and painful mismatch seems to exist between these simplifying assumptions and reality. We argue that effective security in these environments might therefore require rethinking these assumptions.

The Nature of Policy

First, let's explain what exactly we mean by "policy." In the classical way of thinking about computer security, we think about subjects (the entities that do the acting) and objects (the entities that get acted upon); we draw a matrix with rows for each subject and columns for each object and fill in the boxes with permissions: what a given subject is allowed to do with a given object. (Yes, this is a simplifying model.) In many real-world enterprises, these permissions are called entitlements.

In the real world, the subjects are typically real people and the resources those things they must use to get their jobs done. In the computer rendering, these become computer users, programs, and data, governed by some type of access control system. Our community worries about how to craft this system so that it does the right thing—that is, so that it matches the enterprise's real requirements. Hence, we might start by calling out the implicit belief that this goal is in fact possible.

Assumption 1. There exists a correct access control policy for every organization.

Maybe such a policy exists. But maybe it doesn't—or maybe the language we use to render the computerized policy is ineffective at capturing the "it depends" gray areas in the real world. This isn't the real world's fault.

Assumption 2. The correct policy is human-constructible.

Our community gives enterprises an access control system with a set of knobs. Even if a "correct" setting for the knobs exists, what is the tractability of a human organization coming up with it? One infosec officer we know chortled about how computer security researchers believe that it's actually possible for an enterprise to stop what it's doing for two weeks, put everyone in a large room, and work out what the policy is. Even weaker versions of this assumption can be problematic.

Assumption 3. The correct policy is human-recognizable: a human can effectively audit a previous decision.

Assumption 4. The correct policy is human-decidable: a human can say ahead of time whether (user, action, resource) should be allowed.

In the real world, access control questions often lead to the answer "it depends," which requires context. Will the context be available to the parties doing policy creation or auditing? (We heard about a critical control room door that is password-protected—because it needs to be secure—but has the password written on it—because if there's an emergency, someone would need to gain entry.)

Organizational Structures

We now consider assumptions our community has made about human organizations themselves. Secure systems' human components are foreign territory for many computer scientists. Unlike the finite instruction sets that guide the execution of deterministic machines, the principles that govern the action and interaction of people are often beyond any one individual's understanding.

Assumption 5. Resources (and therefore policies) are managed centrally.

Our community typically expects centralized control—but reality often shows that the further we are from the action, the less we understand the real issues. This applies to policy creators, too.

Assumption 6. A corporate organization is structured like a tree, with a few decision-makers at the root and numerous specialist employees at the leaves. Control and decision-making flows in a deterministic manner, one way, along this tree's branches.

At first approximation, modeling the structure of a human organization with a tree is appropriate; management researchers and corporations themselves often use this construction. However, a tree represents the relationships relevant only to the most formal decision-making processes—for example, a tree can help us understand the mechanism by which a university crafts and approves its annual budget. In contrast, in many domains, the decisions most employees make on a day-to-day basis proceed along a more ad hoc path. Influence is determined more by a person's effective job role than their official title, and thus individuals who aren't formally vested with power (for example, administrative assistants) often have a surprising hand in the outcome of small decisions.

The departure from the traditional tree view of an organization is also manifest in larger suborganizational units. For example, a hospital's clinical arm might theoretically be managed by a medical director or other executive to whom individual departments must answer. However, no individual is qualified to directly oversee detailed operations in radiology, neurology, and oncology—each department requires specialist leaders who can make decisions appropriate for the work that department does. This distribution of authority (combined with cultural factors common in a highly educated workforce) makes the organization's process for choosing computer systems or crafting access control policies deviate significantly from the tree-shaped representation.

We've occasionally encountered computer security colleagues surprised by our questioning the assumption of a centralized hierarchy for policy making. We suggest such colleagues contact the business management community, which has been discussing such "matrixed" organization structures for years.

User Knowledge

In other scientific disciplines, when the real world differs from the model, scientists try to fix the model. In our community, we pretend the real world doesn't do that, or we castigate it for misbehaving.

Assumption 7. With sufficient training and commitment to their jobs, well-intentioned users will follow organization policies.

Corollary. Only ill-intentioned users will circumvent control mechanisms.

In the real world, we've repeatedly found users who circumvent the system not because they're evil but because they're conscientiously trying to get their jobs done! One medical clinician even asked us "Are you trying to build a better policeman, or do you want to help patients?" A recent medical journal article[2] provides a wonderful case study of a clever computerized resource-control system—and all the ways that clinicians worked around it to save patient lives.

Assumption 8. The possible negative repercussions of over-entitlement are far greater than the possible negative repercussions of under-entitlement.

Our community touts the principle of "least privilege": a control policy should only grant the minimum access necessary to complete the task at hand. However, this thinking can easily lead to a computerized system that's too restrictive. Perhaps in national security environments, erring on the side of under-entitlement might make sense—but in the domains we've looked at, such errors can result in missed market opportunities or even patient death.

The computer security community laments when passwords are written on sticky notes or under keyboards, but we posit that this phenomenon is often an understandable effort from users to tune a suboptimal access control system. An organization quashes this tuning at its peril; rather than punishing users, we must develop systems and policies that help them do their jobs efficiently without exposing sensitive data to passers-by.

System Knowledge

Can security researchers and practitioners even understand a system? This conundrum is reflected in the complexity of economic systems (and the economic meltdown that experts are still trying to understand and reverse). For humans to be able to manage access control, we implicitly simplify.

Assumption 9. The information relevant to making access control decisions (who has what responsibilities) must change slowly.

In organizations where the tasks are complex and require professional judgment (as for clinicians) or where the nature of the work is highly dynamic (as with investment bankers who are constantly reassigned to new accounts), it might simply be impossible to know ahead of time whether a given operation is going to be acceptable. We must return to rigorous auditing: define the boundaries of how much we trust an individual user, let that person operate broadly within those boundaries, but honestly evaluate what he or she is actually doing. After his initial comments on policy crafting, the infosec officer we alluded to earlier chortled further that, even if he could create a correct policy, we're deluded in thinking that this policy would remain correct for more than a few days.

Assumption 10. Supervisors (or even users) know what entitlements individuals legitimately need.

IT professionals across domains recognize the practice of copy-paste provisioning: a manager, Carla, will admit that when David joins her group, he will clone the entitlements Bob has, given that David's job seems to be similar to Bob's, and Bob gets his done. Neither Carla nor Bob actually know which are the magic combination of permissions, nor are they likely to take valuable time from their actual work to figure it out.

During one organization's effort to reduce over-entitlement, administrators presented users with a list of their entitlements and incentivized them to voluntarily give up ones they deemed unnecessary. The administrators were thrilled when users reduced their permissions up to half—but, of course, dismayed the day when the permission changes went live, and the users couldn't access the data they needed to do their jobs.

The Painful Truth

As disciples of a field that descended in part from mathematics, it's understandable that we don't like the uncertainty that this approach points toward. We like being able to say that a system is "secure." However, pretending that security is a binary property—that it really is possible to sign off on a system or policy as being secure—is driving us to ignore real-world subtleties. We're making a best-guess effort when we make a security policy, but too often we then stick our heads in the sand before we can see what the results are.

As Cureton notes, "policies in and of themselves do not eliminate cybersecurity compromises."[1] If we want to achieve some measure of security, we need to instead observe our systems constantly, acknowledge their complexity, and admit that security is a constant process, not a product of finite action. If our model doesn't match the real world, it's not the real world's fault. □

Acknowledgments

This work is supported in part by the US National Science Foundation's Trustworthy Computing award #0910842.

References

1. L. Cureton, "Our Insecurities, or How to Stop Worrying and Love Compromised Cyber Environments," NASA CIO blog, 5 June 2010; http://wiki.nasa.gov/cm/blog/NASA-CIO-Blog/posts/post_1275770072399.html.
2. R. Koppel et al., "Workarounds to Barcode Medication Administration Systems: Their Occurrences, Causes, and Threats to Patient Safety," *J. Am. Medical Informatics Assoc.*, vol. 15, no. 4, 2008, pp. 408–423.

Sara "Scout" Sinclair *is a PhD candidate at Dartmouth College. Her research interests include access control and usable security. Sinclair has a BA in computer science and French from Wellesley College. When not using a computer, she enjoys crafting and entertaining her pet bird. Contact her at sinclair@cs.dartmouth.edu.*

Sean W. Smith *is an associate professor of computer science at Dartmouth College. His research interests include trusted computing and usable security. Smith has a PhD in computer science from Carnegie Mellon University. His book,* The Craft of System Security *(Addison-Wesley, 2007), is perhaps the only computer security book containing the word "craptastic." Contact him at sws@cs.dartmouth.edu.*