Multidimensional IAM: new trends in enterprise security

Analyst: Wendy Nather

23 May, 2013

One of the complicating factors of identity management is context: it may be straightforward to tell that Jane Doe is who she says she is, but why do we care who she is? This very simple question – why do we care? – means everything when it comes to access management based on this identity.

We care who Jane Doe is because she is John Doe's boss, as attested by the human resources department. We care who she is because she's filling in for Bob this week while he's on vacation – an arrangement that we need to confirm with Bob (and perhaps other authorities, if he's not allowed to delegate certain things). We care because she's in QA and needs to have a wide range of credentials to test an application.

Those are some simple enterprise examples of why identity context matters and why we need authoritative sources to rely on for attributes around them. When you step outside the enterprise, things get even more complicated.

Context implies relationship

It's important to know who Jane Doe is and what roles she plays in case any of them come into conflict. She may work for a healthcare provider, be a patient and be a parent of a patient – all of which require different entitlements, and even different applications. She may not be allowed, as a healthcare worker, to view a patient's records unless she has been validated as the patient's relative and granted access through HIPAA. What's different here is that each role has to be validated from a different source (the employer, the healthcare provider or a form the patient has filled out). More important, it's not just the roles that a person plays within one enterprise that matter: we often need to know what roles they play in other areas of their lives.

With different inputs to validate attributes of an identity, the owner of the IAM system not only

needs to design and establish policies for all of them; it also needs to manage the relationships with any third parties that it will be relying on for these inputs. Depending on how many separate relationships an enterprise has, it may end up managing dozens of attributes per identity, and deciding who will be the authoritative source for each one. This is a particular problem where widespread services are provided and coordinated, such as in the public sector: you might have a combination of juvenile detention, school lunch programs, centers for disease control, K-12 campuses, higher education, nonprofit organizations, educator certification and workforce tracking, all contributing to the same data streams. These all require authoritative sources for the application owner to rely on; there's no way it can manage all of them on its own.

When federation is not enough

In some cases, as with business partners or trusted silos within one organization, federation of identities works perfectly well: if your security model assumes that you will always accept whatever the other party sends you, then it's a great mechanism for integrating systems in a way that is more convenient for everyone.

Delegation is a different matter, though. Delegation is not complete trust; it is limited trust that can be withdrawn at any time. Entities that have regulatory or legal requirements to control access to their systems may not be able to cede full responsibility for identity management in the form of federation. They will always have to keep one hand on the steering wheel, and add their approvals to any access request.

In the case of delegation, it may make sense to allow provisioning by the other party – subject to approval from your side within a workflow. You may need to reserve the right to audit those identities and unilaterally remove their access. As much as possible, partners can do the work of initially validating and setting up those identities, and you may accept their assertion, for example, that someone actually works for them. You also might make it their responsibility to remove identities or change their attributes when they have knowledge of changes in context. But the ultimate responsibility for some additional decision-making rests with you.

The potential complexity of all these business rules is more than many enterprises can handle. How many attributes are needed in order to make an authorization decision around an identity? Where do those attributes come from, and who 'owns' each one? How are they validated, and how often are they audited? What happens when there's an incident involving an identity with attributes managed by multiple organizations? Where are the 'kill switches' for access, and who has the ultimate control of them?

The tangled World Wide Web

Because identity and access management are so closely tied to business rules, an IAM system needs to support the equivalent of a rules engine. At the same time, this complexity needs to be tucked away so that it doesn't overwhelm day-to-day users of the system, especially the non-technical ones. This goes not only for enterprise systems, but for the multi-tenant IAM platforms in use by PaaS, IaaS and SaaS providers. As 451 analyst Steve Coplan wrote in a previous report:

SaaS providers are effectively relying on the enterprise to act as a trusted identity provider, and hinge authorization on internal directory and provisioning logic. The more external services an organization consumes, the more the number of relying parties proliferates. In turn, relying parties are faced with validating a greater number of identity providers and authorizing their users.

These are two different user populations – enterprises and providers – but if you abstract them, the business rules requirements are similar enough that a well-designed system could fulfill both sets. As the Internet grows into a web of peer-to-peer and hierarchical relying parties, IAM offerings will have to keep pace and remain flexible. Now, more than ever, no organization is an island, nor is it flat, nor is it static. Multidimensional IAM is on its way.