AXIOMATICS

White Paper

The Evolution of RBAC Models to Next-Generation ABAC:

An Executive Summary

Overview:

IT security has gone through major changes. Enterprises today are facing a rapid expansion of diverse users, along with an influx of applications, device types, APIs, and microservices, all while navigating new initiatives such as big data and the cloud.

Data is everywhere: we are collecting, storing, and processing it, while needing to both share and protect it—something that becomes increasingly more difficult when data is mission sensitive or business critical. The fact remains, our once simple IT landscape is now a complex, ever-evolving, constantly growing mass of users and the data related to them.

In order to provide secure access, we can no longer afford to define authorization policies based solely on a user's role. We need to know the context surrounding that user, their data, and the interaction between the two. And we need to use that information to provide access to the right user, at the right time, in the right location, and by meeting regulatory compliance. That means evolving an existing Role Based Access Control (RBAC) model to an Attribute Based Access Control (ABAC) model.

ABAC builds upon existing roles, and expands the range of attributes used in the authorization process, while utilizing a context-rich policy language that easily caters to both simple and complex use cases. Equally important, ABAC enables access policies to be defined and enforced within one, manageable central service, instead of being hard-coded into the hundreds or thousands of applications in the typical enterprise's portfolio. As a result, enterprises can now provide well-rounded access control that builds upon RBAC's strengths while harnessing ABAC's ability to capture contextual information.

What is RBAC?

Role Based Access Control (RBAC) was designed for a simpler world. Formalized by NIST in 1992, RBAC quickly became the standard for enterprises managing upwards of 500 employees. Outpacing previous models, RBAC was largely implemented within user provisioning systems, attempting to streamline the joiner, mover, leaver process which gave enterprises the ability to manage access control by role, rather than an employee's individual user ID.

This role provided an extra level of abstraction, acting as a collection of permissions or entitlements. It allows users' access to be based on business functional roles (such as manager or director), or even functional IT roles (such as IT manager or system administrator). And each role could be assigned to one or hundreds of users, making management a far simpler task.

How does Role Based Access Control work?

At its most basic, RBAC works by allowing administrators to assign access permissions or entitlements to roles. Once defined, these roles can be assigned to individual users who may have one or several roles, each with different access rights. And as a business moves forward, new roles can be created and allocated to employees as and when they are required.

Additionally, when employees join the company, change positions within the company, or leave the company, administrators simply have to update their roles via a hierarchical abstraction layer, and assign them (or remove them from) the appropriate roles.

The Limitations of RBAC

But times have changed. Where enterprises once managed thousands of users, now they manage hundreds of thousands. And what once seemed like the perfect solution is showing signs of cracks and limitations. Here are just a few of the ways RBAC has become limited in this new landscape:

Limitation #1: Role Explosion

Because RBAC is limited to defining access permissions by role, an ever-increasing number of users requires an exponentially increasing number of roles in order to accommodate various permission combinations. And because each user often requires entirely unique access rights, one user may be assigned several roles; a one-size-fits-all solution that often results in too much (or too little) access, and potentially an exponential increase in roles vs. users.

Limitation #2: Toxic Combinations

The problem arises from the fact that the various roles assigned to a given user could contain conflicting data. One user could be assigned a role that allows them to create a purchase order, and another that allows them to approve it. These assignments would allow the employee to simultaneously create a purchase order payable to themselves, and then to approve it, posing a significant business risk if not managed properly. In a dynamic authorization system, one that enforces corporate policies, this simply would not be possible.

Limitation #3: Management Nightmares

Between growing numbers of users, and exponentially more roles, role engineering becomes an increasingly difficult task. Administrators have to constantly be on top of changes to users and to roles, and ensure that role assignment combinations are current, accurate, and not conflicting with other roles a user might be assigned. Any attempts to audit or certify such an environment would likewise be fraught with management nightmares. ABAC supports not only dynamic authorization, but also effective auditing of user permissions.

Limitation #4: Lack of Context

Due to the static nature of Role Based Access Control, RBAC is unable to model policies that depend on contextual details such as time-of-day, location, relationship between users, and so on. In other words, RBAC has no way of delineating the relationships between users and using that information to make policy decisions. At its best, RBAC was originally designed to answer just one question: What access does a user have based on their assigned role(s)?

But expanding user populations (partners, consumers, regulators, auditors, etc.), and multi-role users, not to mention rapidly escalating application and database complexity, requires authorization based on a finer level of information. Authorization that not only answers the question, “What is a user’s role(s)?” But also, “Who or what is that user related to?” “What resources should he/she have access to?” “What can they do with this data?” “Where is data being accessed from?” and “At what time?”

These relationships are key to defining safe and secure access control in an ever-changing environment. By defining the context between users and attributes, robust policies can be defined specifying that though a user may create purchase orders, and approve them, he/she is only able to approve purchase orders to which that user is not assigned.

Knowing a user’s role is no longer enough to ensure safe and secure access control. We need context and the relationship information between the various entities involved in the system. We need Attribute Based Access Control (ABAC).

The Future of RBAC

Despite its limitations, there’s no need to reboot and start over. Users are most often the central point around which access control is based. And those users’ roles play an integral part in managing safe and secure access control across an organization. Role Based Access Control provides a solid foundation upon which the future can be built.

But we need to know the context surrounding those users and be able to express complex relational policies at a fine-grained level. Modern data sharing and collaboration scenarios must provide access to the right user, at the right time, in the right location, and by meeting regulatory compliance. By evolving RBAC with ABAC, we can provide well-rounded access control that builds upon RBAC’s strengths with roles while harnessing ABAC’s context for today’s requirements as well as future needs.

The Future of Access Control: ABAC

Keeping What Works from RBAC

As we mentioned, RBAC already has some of the functionality needed to provide safe and secure access control in an evolving environment, and it all starts with a user's role.

Roles are still, and will always be, an integral part of a successful access control strategy. But in order to scale for the future we need to extend those roles using attributes and policies by means of Attribute Based Access Control.

Provisioning User Roles with ABAC Policies: An Anecdote

Mergers and Acquisitions – or affectionately, M&A, are great opportunities for global enterprises to make leaps and bounds in growth. Be it acquiring market share, new technology or new customers, M&A fuels revenues and profit.

The other challenge it presents is merging data – and consumers of that data. And with that comes the challenge of allowing the legacy user base and the new user base to share information, while keeping the most sensitive data locked down as necessary – the balance of which is just not possible with a role-based approach.

Several global enterprises have found that using an Attribute Based Access Control approach to provisioning new users to applications or databases solves this challenge; especially when introducing large populations of new users.

The IAM team at an automobile manufacturer uses the employee role as just one of the many attributes granting access – so with each car brand, confidential data is secured, but global research is shared for the good of the enterprise according to centrally managed policies. They've been able to ward off role explosion – and introduce a truly context-aware approach to fueling competitive advantage.

Several of Axiomatics' global banking customers have been able to quickly absorb branch and geographic acquisitions while keeping customer data secure and visible to only those authorized to service the individual customers. Across industries, transitioning user access from a one-dimensional role-based approach has reduced data leakage and the risk of exposing PII, while excelerating time to market.

Evolving RBAC with Attribute Based Access Control.

Attribute Based Access Control (or ABAC) allows an enterprise to extend existing roles using attributes and policies. By adding context, authorization decisions can be made based not only on a user's role, but also by taking into account who or what that user is related to, what that user needs access to, where that user needs access from, when that user needs access, and how that user is accessing the requested information.

ABAC does this by using policies built upon the individual attributes using natural language. For example, a policy may be written as follows: "Doctors can view medical records of any patient in their department and update any patient record that is directly assigned to them, during working hours and from an approved device." By creating a policy that is easy to understand, with context around a user and what he/she should have access to, access control becomes far more robust.

This functionality expands the scope of RBAC significantly. We no longer need hundreds of overloaded roles, and administrators can add, remove, or reorganize departments and other attributes without having to rewrite the policy. Not to mention, ABAC enables business initiatives not previously available using RBAC alone. For example:

ABAC can be used for delegation: *Doctor Bob: Give Doctor Carol access to my patients when I am away.*

ABAC can be contextually aware: *Doctor Bob can only get access to medical records within a medical context (at the hospital, during office hours from an approved device) and only for specific patients associated with the doctor*

ABAC can be used for collaboration: *The lab researcher: I want to gather as much medical data as I can without breaching patient privacy.*

ABAC can enhance the end-to-end user experience and improve time to get care: *General Practitioners, hospitals, specialist care providers, and insurance companies should be able to work together on the same set of data to improve patient care.*

Evolving RBAC to ABAC with Axiomatics

Though Role Based Access Control has been the dominant access-control method for decades, 21st century business scenarios are stretching its capabilities. Thankfully, ABAC provides a viable alternative that doesn't replace an existing investment in RBAC, but rather extends it, prolonging its usability for the decades to come.

How Axiomatics Evolves RBAC with ABAC

Axiomatics Authorization Solutions were designed to help organizations evolve RBAC using ABAC for fine-grained contextually-aware access control that can scale with an organization across applications, databases and APIs.

Axiomatics Policy Server (APS) is the most complete solution available for enterprise-wide roll out of ABAC, to help begin transitioning from a role-based system to one that can address context of access with attributes and policies. With the flagship Axiomatics dynamic authorization solution, enterprises can roll access to applications under a single point of policy-based management - providing scalability, visibility and control.

Once authorization has been transitioned from RBAC to ABAC, you can also improve protection of information assets at the data layer. The Axiomatics' Data Access Filter uses the same attribute-based approach to help protect content. It dynamically masks or filters data directly in a databases, only allowing users or applications access to the data that they're approved to touch, in accordance with business policies.

Axiomatics solutions easily integrate with new or existing applications and API projects to employ fine-grained authorization that ensures data is protected when needed and shared only when the right circumstances are met.

And no matter your existing infrastructure, Axiomatics Authorization Solutions are compatible with a number of tools that support RBAC including IAM products, API gateways, federated identity solutions, and more.

Next Steps

Contact us to find out how to evolve your RBAC centric system to an ABAC evolved solution. Or, learn more about shifting to an ABAC approach with this Axiomatics executive summary on building a business case for ABAC.

Learn more at: www.axiomatics.com

Trademarks

All brand names and product names used in this book are trademarks, registered trademarks, or trade names of their respective holders.

Copyright © 2016 by Axiomatics AB. All rights reserved.

Our best-in-class software can be deployed across any it environment to safeguard data.
Call or email us to learn how Axiomatics can work with your enterprise:

AXIOMATICS

525 W Monroe St, Suite 2310
Chicago, IL 60661, USA
Tel: +1 (312) 374-3443

Västmannagatan 4
S-111 24 Stockholm, Sweden
Tel: +46 (0)8 51 510 240

info@axiomatics.com www.axiomatics.com twitter.com/axiomatics