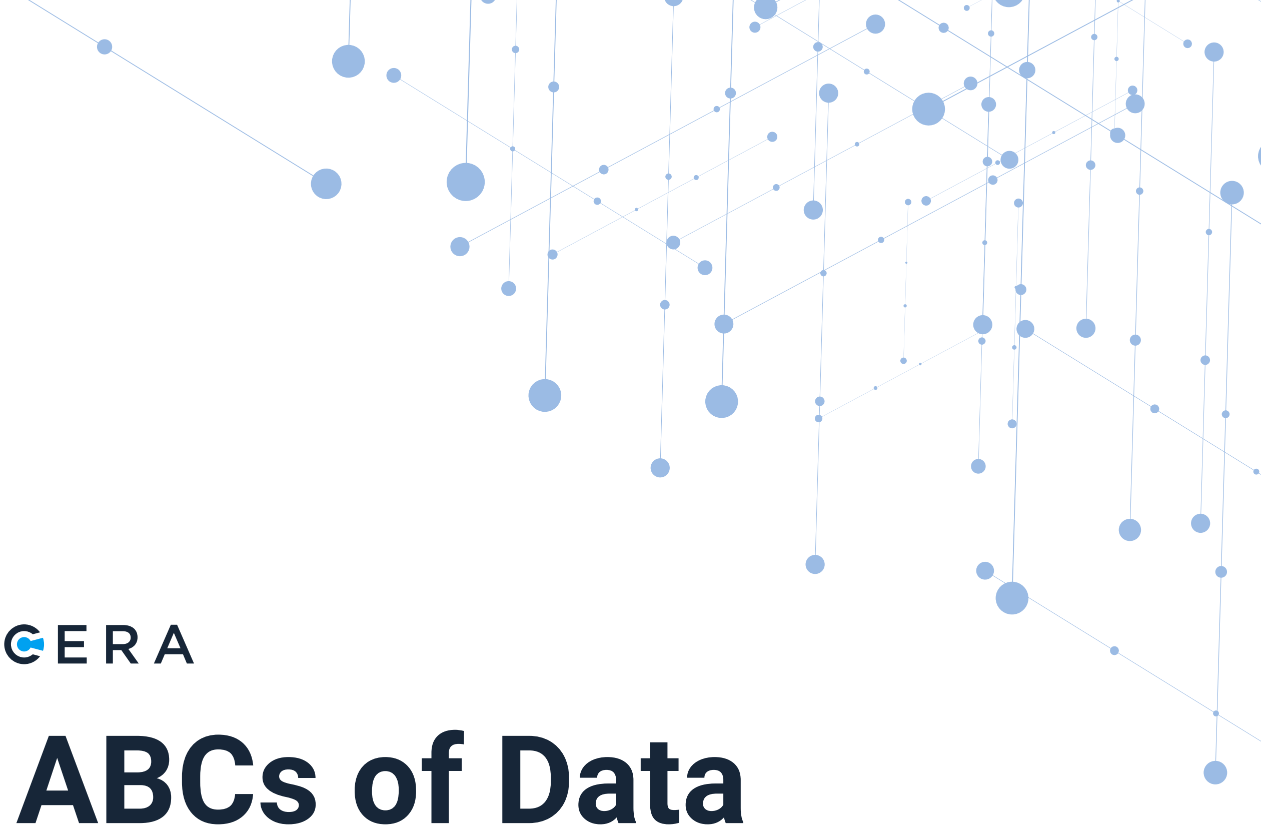


PRIVACERA

The ABCs of Data Access Control

E-BOOK | MARCH 2021

TABLE OF CONTENT

INTRODUCTION

This ebook is a compilation of a series of blogs that provides an overview of foundational concepts of implementing data access controls and policies. The objective of this blog series is to provide:

- Fundamental and objective understanding of the two approaches to administering data access controls: RBAC and ABAC
- Address misconceptions about why companies must choose one approach over other
- Insights that would help companies implement best practices of securing and governing their enterprise data

"Framework – A basic structure underlying a system, concept or text."

CHAPTER 1

Understanding the Element of Access Control

Access control mechanisms have been part of the enterprise IT landscape since the advent of the computer systems.

There are two aspects to controlling access to data. The first aspect relates to authenticating the identity of the user and establishing whether a user or a system is actually who they claim to be. The second aspect has to do with ensuring the user has the appropriate permission to access a system, a process known as authorization.

Mandatory access control (MAC) was one of the first standardized frameworks and is used in systems that require a very high level of security. In MAC a central authority assigns classifications to system resources at an operating system or kernel level. Only users or devices with the required information security clearance can access resources.

The other framework option called discretionary access controls (DAC), involves the owner of a system to define which users or groups have access based on the identity of the subjects and the groups to which they belong. DAC first came out as part of the file permissions (POSIX). This framework is less restrictive than MAC because a subject or a user with a certain permission can advertently or inadvertently share that permission with another subject(s).

An access control list or ACL is a list of permissions related to an object (refer to the definition below). An ACL specifies which subjects or users are granted access to an object, as well as the operations users are allowed to perform on that object. In a filesystem ACL, individual users or group's rights to an object are specified in a table. One may also come across the term active directory (AD), Microsoft's Active Directory Service implemented in an LDAP server, which authenticates and authorizes all users and systems in a Windows domain.

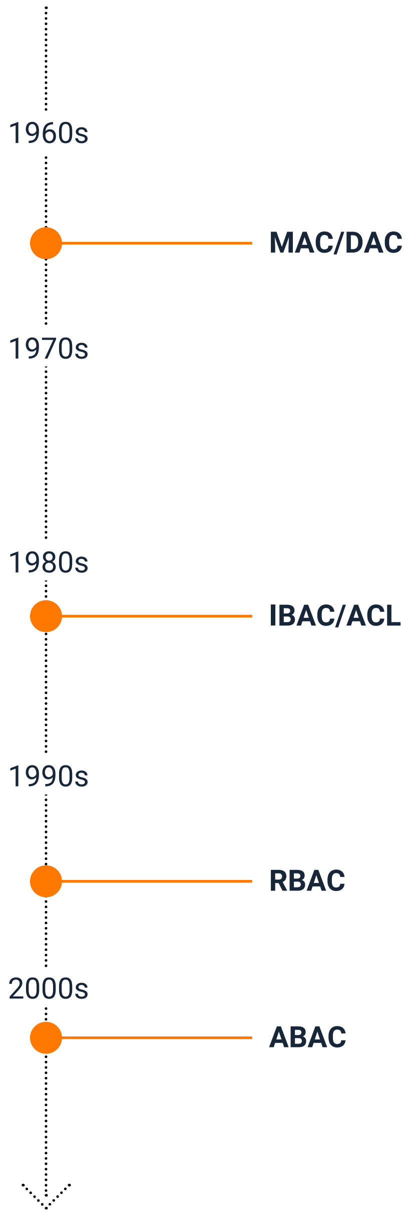


Figure 1: Timeline of various access control mechanisms.

[1] Source: National Institute of Standards and Technology

Today, role-based access control (RBAC) and attribute-based access control (ABAC) are the two most prevalent approaches to managing access to data in the enterprise. RBAC represents an evolution of DAC, wherein certain organizational roles are defined and permissions are granted to those roles. ABAC represents the latest approach to access control and employs users or other attributes as the basis to decide whether to grant access to system resources or not.

The goal of these approaches is to help define the policies and privileges that grant authorized users access to the required data. These users may include data analysts, data scientists, and data engineers. If you are part of a data infrastructure or platform team that is involved in facilitating access to data across the organization, you will likely come across this terminology.

Before we review these two approaches, let's define the baseline terminology that will be helpful in understanding these concepts.

SUBJECT / USER:

Can be an individual, process, or a device that is responsible for performing a task. This task requires the information to be accessed or flowed through the system.

OBJECT:

An object is an information system entity or a data resource on which an operation may be performed. An object can be a database, file, table, row, or column in a table.

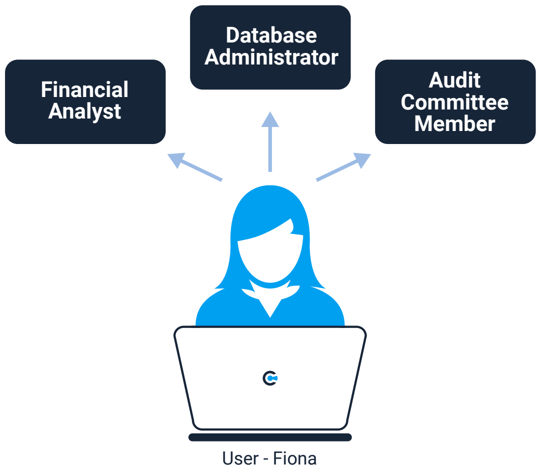


Figure 2: A subject or a user can be assigned to a number of roles

Figure 3: Databases, files and tables are some examples of objects

OPERATION:

An operation is the execution of a function by a subject on an object. In the context of access control, it is the ability to read, write, edit, delete, copy, and modify data. Writing a Select statement against a table is an example of an operation.

ATTRIBUTE:

The characteristic associated with a subject or an object. A subject's location or time zone is an example of his or her attribute.

ROLE:

A set of job functions, duties, or titles associated with certain privileges in the organization.

GROUP:

Users united by a common objective or function. Finance, Marketing, and Engineering departments are examples of groups in an enterprise

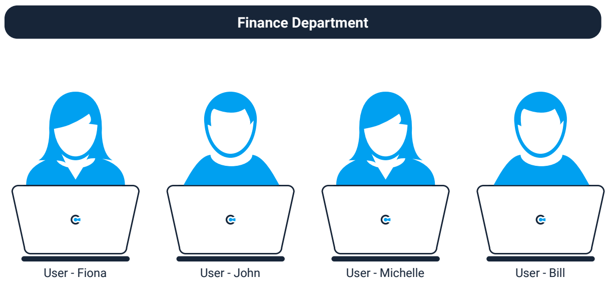


Figure 4: A group has multiple users

PRIVILEGE:

Permission to perform a task or an operation. In this case, it is permission to access a resource, such as data in a database or a table, and perform an operation like read, write, edit, or delete.

In the context of defining data access controls, one may also come across terminology such as administration, authentication, authorization, etc.

- Administration is the centralized management and application of consistent access controls
- Authentication is the ability to establish whether a user or a system is genuine
- Authorization is the ability to provision user access to data with privileges
- Audit is the ability to generate a record of data access at any given time for compliance purposes
- Anonymization is the ability to encrypt data at rest and in motion

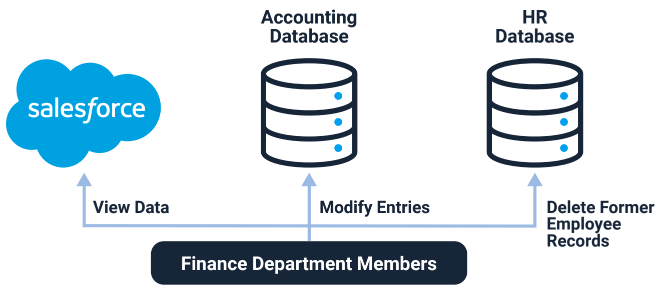


Figure 5: A role can have multiple privileges by resource type

CHAPTER 2

What is RBAC, and Why Should Data Infrastructure Teams Care?

Role Based Access Control's (RBAC) primary objective is to provide a framework that enables enterprises to define and enforce access control policies for their data.

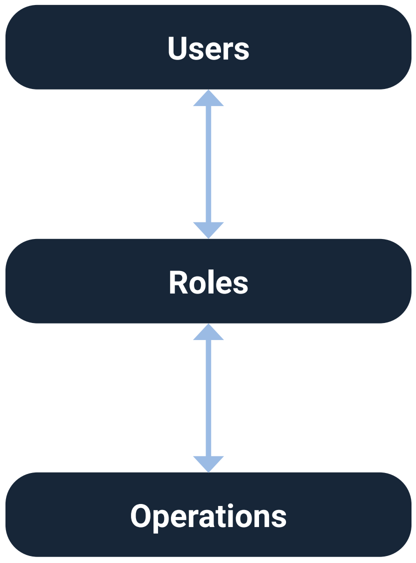
These policies play a critical role in streamlining data security and governance. It is important to note that RBAC's implementation in any enterprise is dictated by its own organizational policies. This provides RBAC the flexibility to be applicable to any organization. As the organizational data access policies evolve over time to meet new requirements or regulations, the controls implemented under RBAC can also be easily modified to reflect those changes.

RBAC is one of the primary approaches to managing access to enterprise data. RBAC is based on the concepts of users, roles, group, and privileges in an organization. We explained these fundamental concepts in the first blog of the series.

In RBAC framework, access to data is associated with roles. Users or subjects are associated with organizational roles. Users can be assigned roles based on their responsibilities or qualifications in the organization. Administrators then assign privileges or permissions to the roles. RBAC simplifies the administration of data access controls, because concepts such as users and roles are well understood constructs in a majority of organizations.

Another advantage of RBAC is the framework's flexibility in enabling administrators to assign users to various roles, reassign users from one role to another, and grant new permissions or revoke permissions as required. Once an RBAC framework is established, the administrator's role is to primarily assign or revoke users to specific roles.

RBAC's framework revolves around the concepts of users, roles, and operations. The relationship between users, roles, and operation is represented in the figure below.



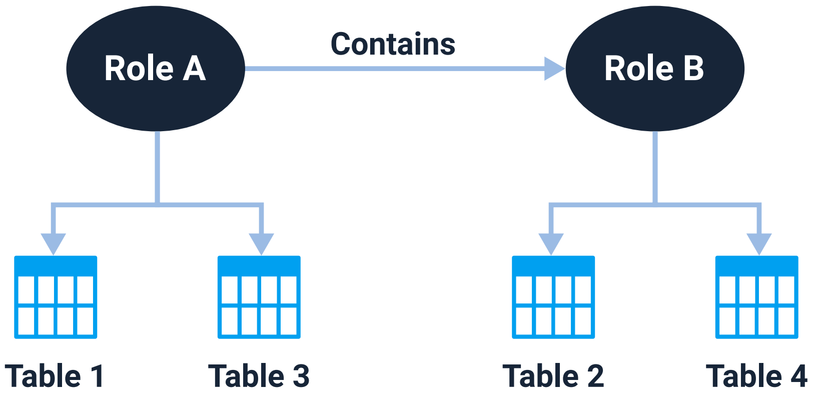
Source: Role-Based Access Control (RBAC): Features and Motivations

There is a many-to-many relationship between users, roles, and operations. This means that a user can be assigned many roles, a role can have many users, and a role or a user can perform many operations. For example, Fiona, who is a financial analyst in an insurance company, is also part of the company's audit committee and is an administrator of the finance department's data mart. Essentially, Fiona has three roles: financial analyst, member of the audit committee, and database administrator. In addition to Fiona, the company's audit committee consists of several members. These include John and Michelle. A single role can also have multiple users – in this case Fiona, John, and Michelle. A role can be granted permission to perform one or more operations. In this case, members of the audit committee may be able to insert, append, or update data in the finance data mart, whereas a user who is not a part of the audit committee will not have these privileges.

> **There is a many-to-many relationship between users, roles, and operations.**

The core RBAC model can be extended by introducing the concept of role hierarchies. This means that a role may be defined in relation to one or more roles. A role hierarchy defines roles that have unique characteristics and that may "contain" other roles. This means that one role may include the permissions, constraints, and objects that are associated with another role. In other words, RBAC accommodates the concept of parent-child relationship between roles. For example, in the figure below, Role A has access to tables 1 and 3, whereas Role B has access to tables 2 and 4. In this case, Role A contains Role B. Based on this definition, the users that have been assigned Role A will also have access to tables 2 and 4, even though they don't have explicit permission to access data in those tables.

In RBAC, administrators also have the ability to define nested relationships where one role can contain other roles, which in turn contain other roles.



Source: [A Revised Model for Role-Based Access Control]

CHAPTER 3

What is ABAC, and Why is it Important?

The concept of Attribute-Based Access Control (ABAC) appeared on the scene in the early 2000s.

Prior to ABAC, managing access to enterprise data involved granting a user or subject permission to perform a specific action on an entity – in this case, a database, table, or column. This was the foundational concept for Role-Based Access Control (RBAC), ABAC's predecessor.

In ABAC framework, granting access or requesting to perform an operation on objects is based on assigned attributes of the subject, object, environment conditions, and a set of policies that are specific to those attributes and conditions[1]. Environment conditions are dynamic factors that are independent of user or object (for example, the time and location of the subject). These conditions may be used as characteristics that contribute to the decision of granting or denying access to an object. Just as subjects or users have attributes, so do objects. Object attributes may include author, creation date, version, effective date, last update, etc.

Object Attributes	Subject Attributes	Environment Conditions Attributes
Type	Name	Location
Author	Employee #	Time Zone
Owner	Designation	Current Time
Date Created	Department/ Affiliation	Current Day
Last Updated	Clearance	
Classification		

Figure 6: Examples of attributes for object, subject and environment conditions[1]

[1] Source: Guide to Attribute Based Access Control (ABAC) Definition and Considerations, NIST Special Publication 800-162, January 2014

DIFFERENCE BETWEEN RBAC & ABAC

In RBAC, access to data is controlled by assigning privileges or permissions to pre-defined organizational roles. Roles are assigned to subjects or users based on their responsibility or area of expertise. For example, a user who is assigned to the role of a manager might have access to a different set of objects and/or given permission to perform a broader set of actions, as compared to a user with an assigned role of an analyst. When a request to access an object is generated by a user, the access control mechanism evaluates the role assigned to the user and the set of operations this role is authorized to perform before making the decision to provision the requested access[1].

ABAC extends the definition of access control beyond the constructs of subject/user, role, and permission to a complex Boolean set of rules that can include a number of diverse attributes. In ABAC, the decision to grant or deny access to requested objects is dependent upon the evaluation of the attributes of subject, object, environment conditions, and the applicable access policy.

ABAC operates by assigning attributes to subjects and objects and developing policies that govern rules of data access. Each object in the information system must be assigned attributes that are specific to the object. Among these attributes, a file can be classified as intellectual property. Similarly, each user or subject in the system must be assigned specific attributes. Among user attributes are location and time zone of the user. Based on these attributes, an administrator can build an access policy that specifies any document classified as intellectual property cannot be accessed by users located outside the U.S., or can only be accessed by users affiliated with the company's legal department between 8 A.M. and 5 P.M. PST.

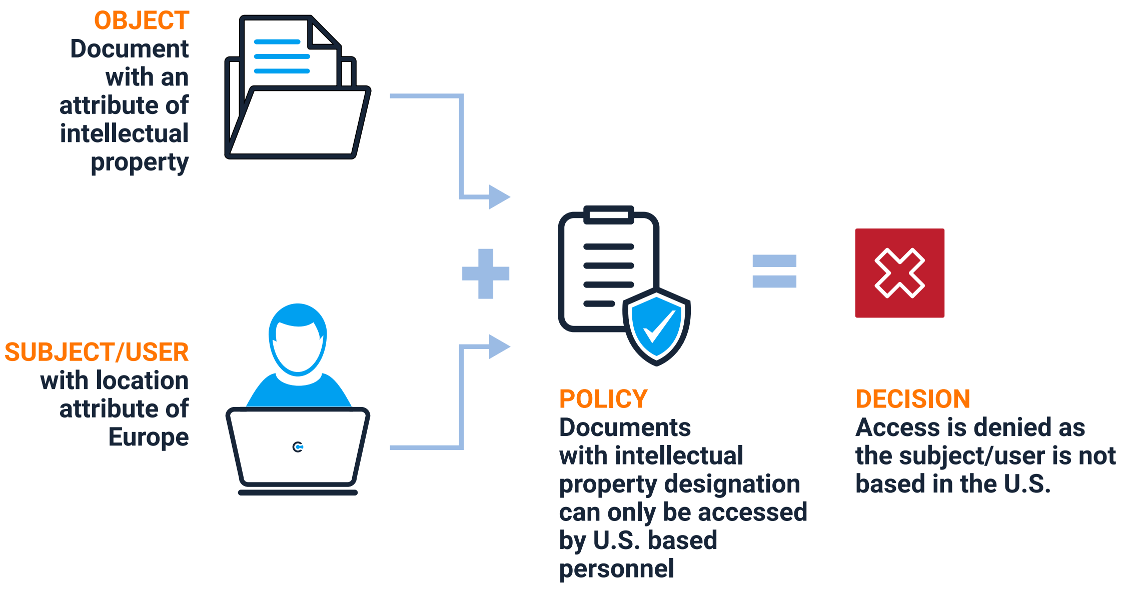

OBJECT
Document with an attribute of intellectual property
SUBJECT/USER
with location attribute of Europe
POLICY
Documents with intellectual property designation can only be accessed by U.S. based personnel
DECISION
Access is denied as the subject/user is not based in the U.S.

Figure 7: In ABAC the combination of object and user attributes governed by a policy results in the decision to grant or deny access to requested object.

[1] Source: Guide to Attribute Based Access Control (ABAC) Definition and Considerations, NIST Special Publication 800-162, January 2014

ABAC offers several advantages to enterprises, as infrastructure administrators do not require knowledge of specific users or subjects who need access to data. The combination of user and object attributes governed by a set of policies can accommodate an unlimited number of users. As new users are added to the platform, they can be governed by the same set of rules; ABAC, therefore, does not require administrators to have prior knowledge of the users, which is why this approach is better suited to environments where users are constantly added and removed from the data platform.

In ABAC, an individual user's attributes can be validated from multiple systems or sources to create a unified identity. Also fine-grained and diverse attributes can be associated with users to address separation of duties and insider threat[1].

CONCLUSION

In conclusion, ABAC provides access to data objects by enforcing policies that are defined based on the attributes of subject and object, operations, and the environment. Generall, ABAC is well suited for large enterprises that are sophisticated enough to enhance or expand their RBAC policies with granular attribute-based policies as these require greater effort to implement and maintain.

“ABAC assigns attributes to subjects and objects to develop policies that govern the rules of data access.”

[1] Source: Guide to Attribute Based Access Control (ABAC) Definition and Considerations, NIST Special Publication 800-162, January 2014

CHAPTER 4

The Evolution of Apache Ranger

As a highly successful open source project used by hundreds of enterprises around the world, Apache Ranger has its roots as a commercial software project.

Balaji Ganesan, Don Bosco Durai, and Selva Neethiraj founded XA Secure in 2013, with the goal to develop an enterprise-ready, centralized platform built from the ground up to define and administer data access controls for on-premises Hadoop data lakes. XA Secure was acquired by Hortonworks in 2014. The company released XA Secure's entire code base, which comprised about 440,000 lines to the Apache Software Foundation (ASF). By making this code widely available, Hortonworks laid the foundation of Apache Ranger as an incubator project, and the first version was released in November 2014. In 2017, ASF recognized Ranger as a top-level project (TLP), a testament to the project adhering to ASF's principles of meritocracy. To date, Ranger has had 15 major and minor releases.

Apache Ranger has been widely adopted by enterprises as an extensible, robust, open source data access control framework that provides comprehensive authorization, audit, and encryption capabilities needed to ffectively govern data in Hadoop-based big data infrastructures. One of the popular misconceptions of Ranger is that it is exclusively based on role-based access control, or RBAC's approach to authorization; however, Ranger's journey as an open source authorization framework began based on ABAC which is a combination of the subject, action, resource, and environment.

Using descriptive attributes of the subjects, resources, and environments–such as AD (active directory) group, Apache Atlas-based tags or classifications, or geo-location — Ranger provides a modern and effective policy approach. ABAC approach is also consistent with recommendations outlined by NIST for ABAC in NIST 800-162. This approach enables compliance personnel and security administrators to define precise and intuitive security policies at a very fine-grained level for each resource.

Another common misconception of Ranger is that it follows a rejection-based approach to access control. In other words, administrators must explicitly specify which users are restricted to access data resources, which is inefficient and counter-intuitive. The reality is that Ranger follows the industry best practice of creating data access policies with the least privilege, under which users are explicitly denied access, unless there is a specific policy in place to grant them access. For example, a user may only have permission to "select," or view data from a table, but not "update" or modify data.

Ranger further enhances this best practice by prioritizing "deny" conditions to supersede "allow" conditions by default. The ability to support conditions for deny/allow, along with specific exclude/include conditions, enables security and compliance administrators to achieve truly fine-grained access control by writing a small set of easily understandable policies. In some cases, what would have required a dozen roles and permissions to specify a policy, can now be done with a single, simple policy in Apache Ranger's robust policy framework.

Privacera has stayed true to its open source heritage by extending Ranger's industry best practices to cloud services. For companies in the midst of digital transformations by migrating data and analytical workloads to the cloud, Privacera provides a faster path to onboard data analysts and scientists by reusing the same robust Ranger policies in cloud that are used to govern data in on-premises data lakes.

"Over time, Ranger encompassed both ABAC and RBAC approaches to provide a holistic approach to data governance."

CHAPTER 5

RBAC vs. ABAC – Is that the Right Debate?

As companies migrate their analytic workloads to the cloud in shared environments, there is an active debate between the various approaches to implementing data access controls – specifically whether ABAC offers superior security and governance than RBAC.

Because modern data infrastructures are becoming increasingly sophisticated and complex, the effectiveness of IT solutions cannot be based on just one criterion – leaving the ABAC versus RBAC debate misguided for many practitioners.

WHICH DATA ACCESS CONTROL SOLUTION IS THE BEST?

The answer to this question depends on which approach is most efficient for your business. Various solutions currently on the market can only be compared by their ability to strike a balance between empowering administrators to make data widely available in their organizations and complying with industry and privacy regulations, without adversely affecting the performance of the data platform.

Organizations' use cases, existing infrastructure, experience with various cloud environments, and the user learning curve are only a few variables that make the RBAC vs. ABAC debate one dimensional. Companies in the middle of a digital transformation project are less concerned about whether the solution is based on role or attribute-based controls and more concerned with how quickly they can migrate existing data and access control policies to the cloud.

But let's start with the current state of the market and a few misconceptions.The majority of companies migrating to the cloud are implementing hybrid architectures. These companies either have an on-premises implementation of a traditional database, like Teradata, DB2 Oracle, etc., or they have a Hadoop-based data lake they want to migrate to the cloud to take advantage of reduced operational burden, increased agility, and elasticity.

It may not be common knowledge that Apache Ranger is the original ABAC solution for heterogeneous data services and this blog published by the Apache Software Foundation in 2017 attests to this fact. In fact, Ranger started its journey as an open source, ABAC solution. In addition to empowering data administrators to define access policies based on roles and users, Ranger also offers the flexibility to authorize policies based on a combination of subject, action, resource, and environment. Using descriptive attributes such as AD group, Apache Atlas-based tags or classifications, and geo-location, Ranger provides a holistic approach to data governance that encompasses both ABAC and RBAC approaches.

DATA ACCESS CONTROL MUST EASILY INTEGRATE WITH EXISTING BUSINESS PROCESSES

As data accumulates, it becomes increasingly difficult to move; this is also true for data access policies. Companies invest significant effort and resources to build unique access control policies over a long period of time. In fact, these access policies are a part of, and reflect, they way they do business. Therefore, it is unreasonable to expect companies migrating to the cloud to drop all of their existing data governance policies and start from scratch to recreate them in the cloud. To the contrary, when we speak with customers, they routinely ask how to leverage their on-premises data lake access policies for their cloud environments.

CAN YOUR CHOICE OF ACCESS CONTROL ACCELERATE SECURE DATA SHARING?

A more practical way for data administrators to think about this debate is to determine if the solution they are using helps accelerate access to data for data analysts and data scientists with proper controls in place.

Among the data lakes built by utilizing big data vendors like Cloudera, Hortonworks, and MapR, Apache Ranger has been the predominant mechanism of administering access control. Ranger has been effectively deployed at thousands of companies to define, authorize, and administer access control policies from a single-pane across various open source compute engines, including: Apache Spark, Apache Hive, and Apache Kafka.

CONCLUSION

According to a Mckinsey survey[1], 69% of organizations cite that "implementing stringent security guidelines and code review processes can slow developers significantly." This is especially troubling, because one of the primary drivers of cloud migration is the speed and agility of data access and analysis.

To avoid delaying data access to data analysts and scientists, rewriting data access policies from scratch, or retraining users on a new platform, companies should use their existing Ranger policies in the cloud. By doing so, they will enable faster user onboarding; fast, secure access to data for analysis; and most importantly, faster access to the benefits of digital transformation.

"As data accumulates, it becomes increasingly difficult to move; this is also true for data access policies."

[1] Source: Unlocking business acceleration in a hybrid cloud world, McKinsey Digital, Aug 5th, 2019.

PRIVACERA

39300 Civic Center Drive
Suite 140
Fremont, CA 94538

privacera.com

questions@privacera.com
510-413-7300

privacera @privacera