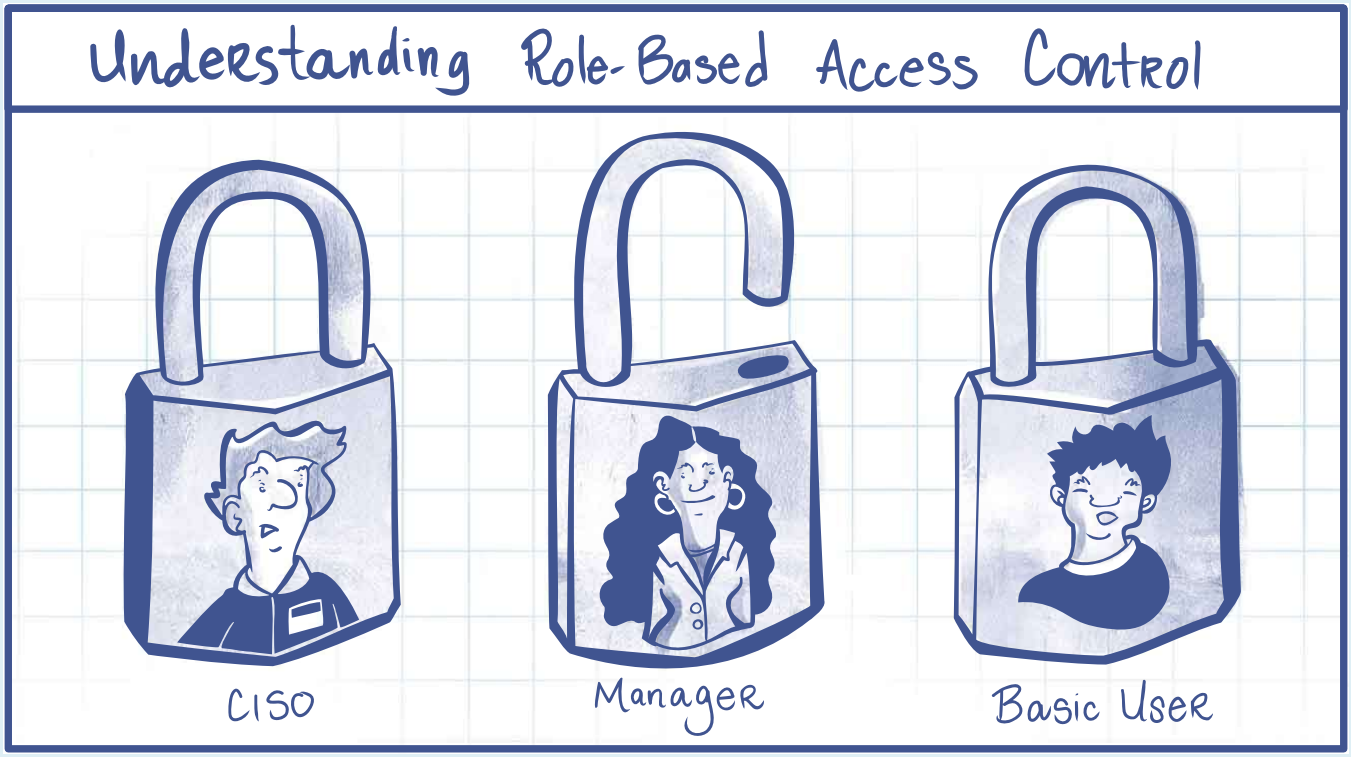
Understanding Role-Based Access Control (RBAC)

strongdm

Table of Contents



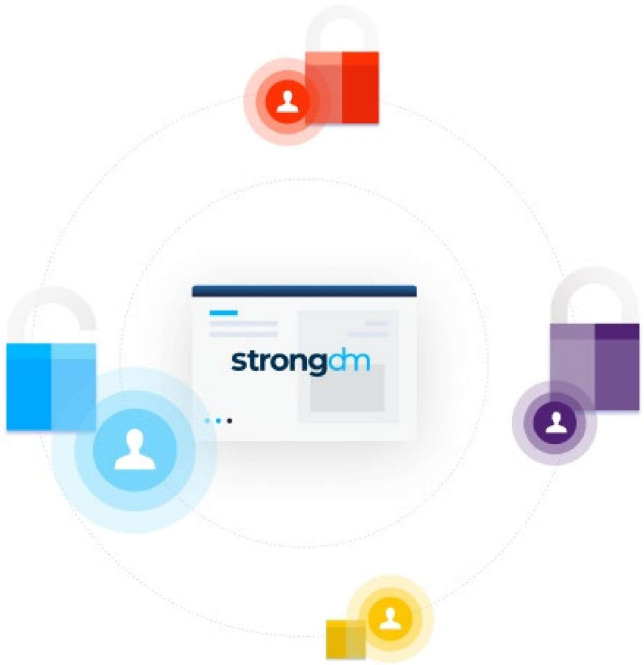
TL;DR

TL;DR: This article provides a comprehensive overview of role-based access control (RBAC) along with a guided approach to implementing, maintaining, and extending RBAC to suit the needs of your organization. You will discover what roles are, how to define them, and how using them to govern access can help secure your network, reduce administrative overhead, and help you achieve regulatory compliance. Let's jump in.

01

What is RBAC?

Role-based access control (RBAC) is a security approach that authorizes and restricts system access to users based on their role(s) within an organization. This allows users to access the data and applications needed to fulfill their job requirements and minimizes the risk of unauthorized employees accessing sensitive information or performing unauthorized tasks. In addition to restricting access, RBAC can refine the way a user interacts with data—permitting read-only or read/write access to certain roles, thus limiting a user's ability to execute commands or delete information.



An effective system of user access control is essential for large enterprises or companies that manage a large number of contractors, vendors, and even customers. For these organizations, RBAC will protect critical data, improve operational efficiency, and help certify regulatory compliance. We'll cover all of this later in the article. But first, let's take a quick look at where it all began.

02

History of Role-based Access Control

People have used roles and responsibilities to moderate access to commercial computer systems since at least the 1970s. However, these procedures were ad hoc and often had to be redesigned on a per-case basis for each new system.

It wasn't until 1992 that researchers at the American National Standards Institute (NIST) first began to formalize the system we know as role-based access control. In that year, Ferraiolo and Kuhn laid the foundation for the model we use today in a paper outlining a general-purpose access control methodology appropriate for civilian and commercial use.

Ferraiolo, Kuhn, and others continued to refine RBAC throughout the 1990s and early 2000s, building on earlier work to explore the economic benefits of RBAC, outline a unified model, and most notably, define separation of duty forms. In 2004, NIST officially adopted RBAC as an industry standard.

03

The RBAC Model

People have used roles and responsibilities to moderate access to commercial computer systems since at least the 1970s. However, these procedures were ad hoc and often had to be redesigned on a per-case basis for each new system.

Type	Detail
Core RBAC	The core model outlines the essential elements of every role-based access control system. While core RBAC can stand alone as an access control method, it also lays the foundation for both the hierarchical and constrained models. As such, all RBAC must adhere to the following three rules: • Role assignment: A subject can exercise a permission only if the subject has selected or been assigned a role. • Role authorization: A subject's active role must be authorized. • Permission authorization: A subject can only exercise a permission which is authorized for the subject's active role.
Hierarchical RBAC	Hierarchical RBAC builds on the core model by establishing inheritance between roles. In this model, elevated users acquire the permission of all roles below them. The needs of the organization determine how complex the hierarchy must be.
Constrained RBAC	This third RBAC standard adds separation of duties to the core model. Separation of duty relations fall under two headings: static and dynamic. • Under static separation of duty relations (SSD), a single user cannot hold mutually-exclusive roles (as defined by the organization). This ensures, for example, that one individual cannot both make and approve a purchase. • In the Dynamic Separation of Duty (DSD) model, a user can be a member of conflicting roles. However the user may not function in both roles during a single session. This constraint helps control internal security threats by, for example, enforcing the two-person rule in which two distinct users are required to authorize an action.

04

Examples of Role-based Access Control

To see how this might work, consider an analogy (oversimplified for the sake of clarity).

Steve oversees a youth hockey team composed of players, coaches, referees, and a concession seller. Even in this very basic organization, complex rights and responsibilities exist. Players need access to locker rooms and the ice rink, while the referee needs access only to the ice. The coach must create and share a roster of players, including contact information. The person selling concessions interacts with items for sale, processes credit cards, and collects/deposits cash. The administrator organizes the game schedule, gathers fees from players, pays coaches and refs (or organizes volunteers). And so on.

In this scenario, allocating access according to roles is a useful approach. When a new child joins the team, she is issued a player uniform and a read-only copy of the roster. She is given a game and practice schedule and the combination to her locker. Nobody has to consider whether she needs a key to the cash register or a pair of referee's orange armbands.

The same principle can apply to businesses and other institutions. Here are a few examples of RBAC in action:

- An HR manager at a large enterprise is authorized to add, change, or delete employee records for anyone in the organization (except herself, see separation of duties) but has no access to the customer database.
- A member of the sales team at a medium-sized IT vendor is assigned a role that allows him to update the customer database, but he is unable to view the details of employee records other than his own.
- A doctor at a large hospital has access to complete patient records, while the receptionist can access only basic contact information.

05

Benefits of Role-based Access Control

As you can see from these examples, RBAC provides a layer of security without interrupting peoples' ability to do their jobs. Here are the three primary benefits of role-based access control.

Type	Detail
Increases Security	RBAC restricts user access to the minimum levels required to perform a job. This helps organizations enforce security best-practices like the principle of least privilege (PoLP), which diminishes the risk of data breaches and data leakage. Should a breach occur, RBAC limits the impact by shrinking the attack surface—access to protected information will be limited to the role that the hacker used as an entry point. So, for example, an HR employee who falls victim to a phishing attack cannot expose privileged information from the finance department. Malicious attacks on any single account are stifled before they can harm other systems.

Type	Detail
Simplifies Workflows	RBAC grants users the exact access needed for their roles, which helps eliminate the bottlenecks. Employees no longer have to badger admins for access to data and systems. And IT is freed from the mountain of busywork required to manage one-off permission for each user. RBAC simplifies onboarding, offboarding, and other provisioning/deprovisioning activities. Admins can easily update permissions for existing employees who change roles within the organization or for contractors and third-party users who need temporary access to your network. These increases in operational efficiency offer economic benefits and improve employee satisfaction. A win-win.
Improves Compliance	Finally, RBAC improves your compliance posture. All organizations—from health care providers and IT vendors to financial institutions—must adhere to federal, state, and local regulations for privacy and confidentiality. Additionally, compliance certifications like SOC 2 can improve brand reputation and offer a competitive advantage for businesses that regularly handle third-party data. Demonstrating compliance reflects a commitment to ensuring the security of customer data as well as the ability to protect sensitive information in general. RBAC provides a framework for managing and monitoring access. Administrators know who accessed a system and when, what modifications were made, and what authorizations were in effect. This helps organizations correct any issues that arise and makes it easier to meet regulatory requirements such as HIPAA, SOX, SOC 2, and ISO 27001, which depend on network visibility to prove that data and sensitive information have been handled according to privacy, security, and confidentiality standards.

05

Disadvantages of Role-based Access Control

For many users, the standard definition of RBAC as a method of restricting access conjures images of frustration and friction, and without thoughtful implementation, those outcomes may indeed come to fruition. Later in this article, we will outline how to set up and manage RBAC smoothly. But first, here are four potential disadvantages of the model.

Type	Detail
Requires Business Knowledge	There is no one-size-fits-all approach to defining roles. Organizations must coordinate across departments when determining how to categorize roles and how to manage access for those roles. This requires a clear understanding of the ideal structure of the organization as well as the technical infrastructure that supports it. In large or growing organizations, this can be a daunting task made harder when IT or security managers are expected to define roles without the help of HR or executive decision-makers. This common attempt to simplify implementation actually makes the problem worse, resulting in a misalignment with larger company goals.
Demands Thoughtful Implementation	Assigning roles can be a challenge. Is it ever the case that junior staff members need more access than their managers, or is a hierarchical structure more important? Do security teams need access to the data they are trying to secure and at what level (create/read/update/delete)? Should a user be assigned a role outside of their department in order to secure temporary access to privileged files? Numerous questions may arise, and the answers won't always be clear.
Lacks Flexibility	RBAC has a reputation for being too rigid, and it's no wonder. Organizations grow, teams expand, and access needs shift. Roles you defined at the beginning of your RBAC project may no longer fit company goals. On top of that, admins face pressure to quickly onboard new hires—even those with as-yet undefined roles. The result? People's roles and level of permissions may be misaligned. For instance, a person may be given too many permissions for his or her role, assigned to too many roles, or perhaps a combination of both.. While these efforts may work as a quick fix, they also create security gaps and compliance challenges—undoing the whole reason you implemented RBAC in the first place!
Leads to Role Explosion	Some teams attempt to side-step the problems above by defining increasingly fine-grained roles, creating ad hoc roles as new needs emerge, or assigning too many roles to individual users. While this may ease friction in the short term, it also makes RBAC confusing and difficult to manage. This problem, often termed role explosion, is one of the most common objections to RBAC. It arises when real-world roles and access needs differ from those outlined in your policy documents, even in very minor ways. And roles created as a temporary fix have a habit of sticking around. Admins may forget or even purposely choose to leave these roles in place, even when the people for whom they were created leave or change jobs within the organization. The result: privilege creep and chaos.

06

RBAC vs. ABAC vs. ACL vs. PBAC

Effective access control is any method that secures your network without creating friction for users. While RBAC remains a popular approach to restricting access, you may want to consider other options. attribute-based access control (ABAC), access control lists (ACL), and policy-based access control are three alternatives. Let's take a look at each in turn.

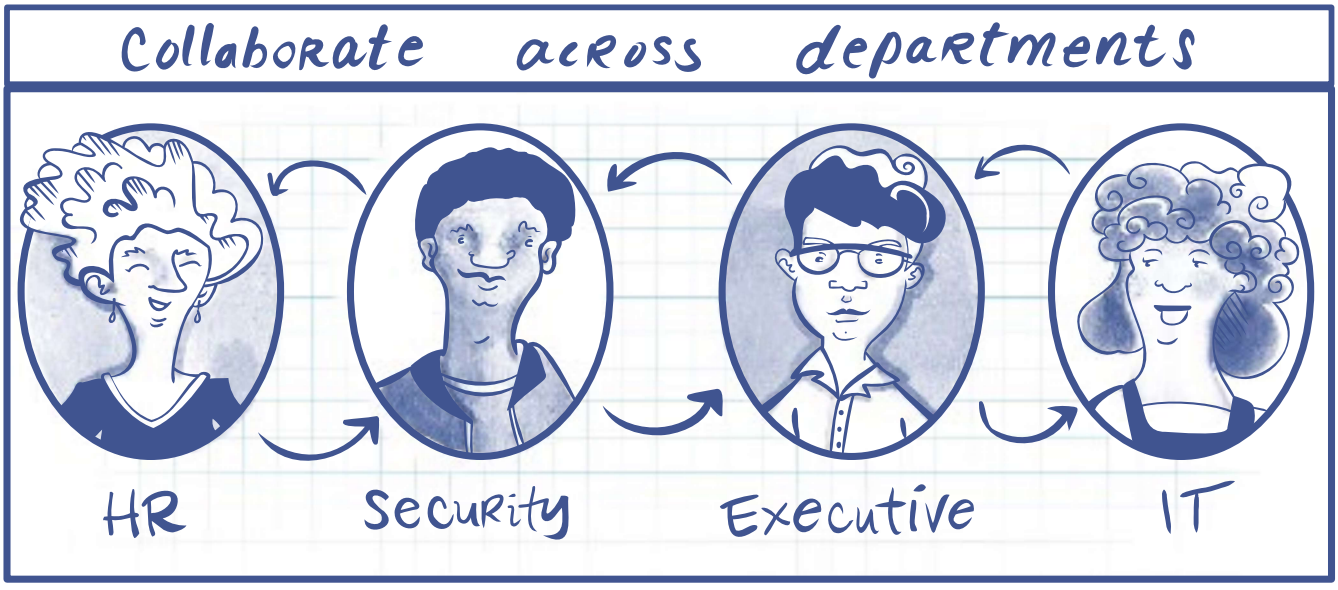
Type	Detail
RBAC vs. ABAC	Organizations use ABAC to achieve more fine-grained access control—either as a replacement for or supplement to RBAC. Unlike RBAC, which grants access according to predefined roles, ABAC relies on a combination of attributes to match users with the resources they need to do a job. Attributes may include: • user demographics such as name, organization, job title, or security clearance. • resource properties such as owner or creation date. • environmental specifics such as time of day, location of access, and threat levels. By evaluating attributes of both subjects and resources, ABAC allows for more flexibility in policy creation and enforcement. With ABAC, access is dynamic. Decisions can be made according to context and risk at runtime. ABAC policies can be enforced on servers, databases, clusters, and a host of other resources. However, granularity introduces complexity, which can create implementation hassles and extra administrative busywork for IT. Additionally, ABAC may simply replace role explosion with a new problem—attribute accretion.
RBAC vs. ACL	Access control lists (ACL) control or restrict the flow of traffic through a digital environment. ACL rules grant or deny access in two general categories: • Filesystem ACLs apply to files and/or directories. The ACL specifies which subject (human user or machine/system process) is allowed access to objects and what operations are allowed on those objects. • Networking ACLs apply to the network routers and switches. The ACL specifies which type of traffic can access the network and what activity is allowed. Organizations use ACLs in tandem with VPNs to manage traffic. Doing so may improve network performance, increase security, and allow for more granular monitoring at entry and exit points. This makes ACLs suitable for securing individual users and low-level data, but it may not be the best approach to access management in most business applications.

Type	Detail
RBAC vs. PBAC	Policy-Based Access Control (PBAC) is another access management strategy that focuses on authorization. Whereas RBAC restricts user access based on static roles, PBAC determines access privileges dynamically based on rules and policies. Although PBAC is fairly similar to ABAC, ABAC requires more IT and development resources (e.g., XML coding) as the number of attributes required increases. Some key benefits of PBAC include: • flexibility to be fine- or coarse-grained. • ability to quickly add, remove, or amend permissions. • environmental and contextual controls (e.g., time- or location-bound access). • visibility into the relationship between identities and resources. These benefits may make access control more nimble, especially as organizations grow and change. But every access control system will require some amount of management and thoughtful implementation. Luckily, some advanced planning and a little elbow grease can make the task productive, successful, and even fun. Here's how.

07

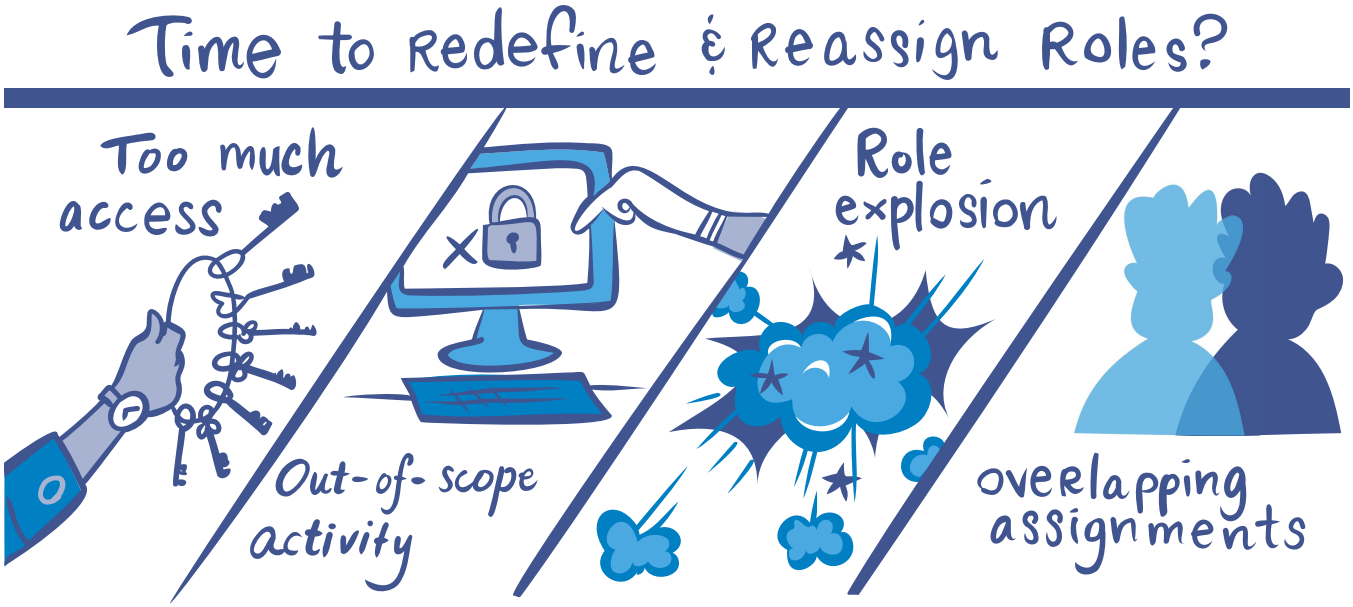
How to Implement Access Control | RBAC Best Practices

Set yourself up for success by following these role-based access control best practices. Don't expect IT to implement RBAC alone. Begin with a conversation across departments, and then proceed systematically to ease the transition and avoid unnecessary friction in your workforce as new systems roll out. RBAC implementation requires high-level understanding of business structure and goals. By collaborating from the start, you will be better prepared to reap the benefits of RBAC and get the most out of your efforts.



Type	Detail
Develop an RBAC Strategy	Start by evaluating where you are. What systems, data, or processes in your organization would benefit from access control? Be sure to include any job functions, technologies, and business operations. Paint with broad strokes in the beginning. You will refine the process as you go along. Next, consider where you want to be. Will you use RBAC to automate provisioning? Do you need a better way to control access to applications that store sensitive data? What is your desired outcome for this process? Finally, note any gaps you need to tackle. Are your authentication/ authorization models consistent across your organization? Are there compliance or regulatory requirements you need to meet? Was there a security event that prompted you to switch to RBAC? Once you have mapped out your strategy, you are ready to move on to the details.
Inventory Your Systems	Make a list of every resource or service that requires access control. The list may include email, cloud apps, customer databases, shared folders on a file server, and so on..
Analyze Your Workforce	Role and access discovery is both art and science, and collaboration across IT, HR, and executive leaders will make the process easier. Start by grouping your workforce into roles based on shared access needs. Be sure to include both current and planned departments. At the same time, avoid the trap of defining too many roles. How many is too many? That will depend on your organization. The right number will restrict access enough to secure your systems without stifling creativity. Larger organizations may require a more systematic method of role creation in order to avoid common pitfalls such as role explosion, role overlap, and over-reliance on exceptions. Here, we recommend a two-pronged approach. • Evaluate roles from the top down. Business managers should design a set of roles that align with company goals and take the entire workforce into consideration. Rather than focusing on systems and technology, the top-down approach should address the functional access needs of each role. • Concurrent with phase one, IT can begin a bottom-up analysis, gathering information about the way users are accessing systems. Then generate roles based on this analysis.

Type	Detail
Create and Define Roles	Finally, reconcile your lists. Map the result of your workforce analysis to the resources from your inventory according to the principle of least privilege. This mapping will define your roles. For example, you may create a Basic User role, which has access to email and Slack and applies to all users in the organization. You may create a specialist role, such as Hiring Manager, which has read/write access to the employee database. You may create an Employee Database Administrator role, which has full control of the employee database. And so on for each department.
Establish a Governance Structure	In addition to defining roles, you need to establish a decision-making body to maintain them. Articulate, in writing, the project priorities and standards that serve the best interest of your organization as a whole. Your access control policies may include: • performance measures • risk-management strategies • role re-evaluation guidelines • direction regarding who maintains roles • a plan to keep the policy up to date Policy-based access control helps prevent role proliferation and keeps your RBAC project on track even as your company grows or conflicts between departments arise.
Assign People to Roles	All of that preparation has laid the groundwork for the final step: implementation. Now that you have inventoried your systems and outlined the way your workforce uses them, it is time to assign roles to your employees and begin using RBAC to manage access rights and permissions. Larger organizations may choose to roll out RBAC in stages. Start with a small group of users, organized around a business function or department. Collect feedback and make any adjustments before moving to the next stage. This will minimize workforce disruption, help you build on small successes, and demonstrate the value of the role-based access control model. Congratulations! You have now successfully implemented RBAC in your organization. Your next task is to keep it running smoothly.



08

Managing Role-Based Access Control

Needs change. Systems change. People come and go. The RBAC you design at the start of this project will invariably differ from the RBAC you need down the road. In the early days of implementation, keep an eye on your security status and fine-tune your roles as needed. Once you reach stability, settle into a consistent cadence of regular review—annually or quarterly depending on the needs of your organization.

Using roles simplifies the process of adding, removing, and adjusting permissions to individual users, but as your organization grows in complexity, you will need to modify your roles as well. This is where iterative adjustment and regular review come into play.

Continue to collect feedback and monitor your security status on an ongoing basis. And conduct a periodic review of roles, role assignments, and RBAC authorization. Review access logs and user feedback to discover what's working and what may need to change.

Keep an eye out for:

- roles with unnecessary access to a particular resource.
- users attempting to access data outside the scope of their role.
- overlapping role assignments.
- role proliferation/role explosion.

These could be signs that you need to redefine and reassign roles. It may also be a good time to educate your workforce about security best practices. However, if you have taken all of these steps and still struggle to control access, never fear. RBAC has a few more tricks up its sleeves.

09

Extending RBAC | A Rule-based Approach

Teams managing a large volume of resources, especially ephemeral ones, need more flexible access rules to keep up with their ever-changing infrastructure. But this doesn't mean scrapping RBAC and starting over.

Rather than replacing role-based access control, you can complement it with ABAC policies. With the flexibility that dynamic access control affords, you now are in position to scale your organization without sacrificing any prior investments made in your RBAC system.

Extending RBAC in this way makes it more nimble, preventing the bottlenecks and workflow disruption that follow from delays in provisioning, while maintaining the security benefit of timely revocation of unauthorized access.

Managing access in the modern cloud presents new challenges that must be met with a collaborative approach that balances network security with workforce happiness. Role-based access control simplifies user provisioning, helps satisfy audit requirements, and facilitates adherence to security best-practices such as PoLP and Zero Trust. So give it a try. The effort you put in today will yield great benefits in the long run.

Unsure where to start? We've got your RBAC! Visit strongDM, and sign up for our free, no-BS demo today.

10

More RBAC Resources

Free Role and Access Discovery Workbook

Kubernetes Role-Based Access Control (RBAC)

Access Management 101: Understanding Roles & Access

Getting Started: Role & Access Discovery

Role & Access Discovery | Who Has Access to What Now?

strongDM's infrastructure access platform gives every business secure access controls in a way folks love to use. Trusted by the Fortune 500 to fast-growing businesses like Peloton, SoFi, Chime, Yext, and Better, strongDM gives businesses the control and visibility they need at the speed they want with one platform that works for every environment. strongDM is intentionally distributed. Head to **www.strongdm.com** to learn more.