APONO

A Buyer's Guide to Privileged Access Governance Solutions

18 pages

Introduction

Most companies in the world today have already migrated most of their workloads to the cloud, with 91 percent of organizations reporting that they already have or will refactor their applications to use cloud-native technologies.

The major cloud providers, AWS, Azure, and GCP, are now the way that most people work. And most resources, such as databases or machines, are running in the cloud. Yet few teams can effectively manage identities in the cloud at scale, with Gartner estimating that by 2023, 75 percent of cloud security failures will occur due to inadequate management of identities and accesses.

As a result, controlling, monitoring and auditing privileged access has become even more critical for protecting against both external and internal threat vectors, human error and a growing list of compliance requirements. However, organizations are growing increasingly frustrated with the existing solutions designed to deal with the unique nature of privileged resources, leading to a new category called Privileged Access Governance (PAG).

In this guide, we examine (among other things) the following:

- Where existing PAM solutions fall short
- The emergence of a new privileged access solution, Privileged Access Governance
- The 10 questions you should ask when evaluating a Privileged Access Governance solution

Why you need Privileged Access in the Cloud

A whopping 94 percent of enterprises report they are using cloud services today, and 75 percent say security is a top concern.

01. Agility

Agile development has created a world where the environment is changing on an hourly basis as organizations push new code to production and create new cloud instances all the time. With that, access to support customers, fix bugs, and do production maintenance is required more often.

In addition, it's not just IT teams that manage access to different systems, but also DevOps and the engineers themselves that need to have a deep understanding and strong capabilities in each new cloud, app and service.

WHY YOU NEED PRIVILEGED ACCESS IN THE CLOUD

02. Scale

A third of enterprises spend at least $12 million annually on the public cloud, which translates to huge cloud environments. In addition, 92 percent of organizations use at least two clouds, as multi-cloud is becoming the leading approach.

This means more access to manage, with new environments, services and apps being spun up all the time. AWS alone has a whopping 200 cloud services, and a real cloud environment can have tens of thousands of instances for each one. It's harder than ever for the business to keep up, let alone manage access among so many cloud providers, services, instances, humans and machines.

03. Regulatory Compliance

Stricter regulations make it more complex to manage access. Regulatory bodies and industry standards are placing greater emphasis on the need to control and monitor privileged access.

Compliance frameworks like GDPR, HIPAA, and PCI-DSS require organizations to implement measures to ensure that only authorized personnel can access sensitive data, and most tech vendors today must also comply with SOC2 and other voluntary standards that enable business.

04. Auditing and Accountability

Privileged access solutions often provide auditing and reporting capabilities.

This is crucial for demonstrating compliance, conducting post-incident analysis, and maintaining accountability for privileged access activities.

05. Rising Cybersecurity Threats

Cybersecurity threats, including data breaches, ransomware attacks, and insider threats, have been on the rise.

Attackers often target privileged accounts because they provide them with the highest level of access and control within an organization's IT infrastructure. Proper PAG helps to mitigate the risks associated with unauthorized access to sensitive systems and data.

What is Privileged Access Management (PAM)?

Gartner defines Privileged Access Management (PAM) as tools that manage and protect accounts, credentials and commands that offer an elevated level of technical access, that is, administer or configure systems and applications.

Available as software, SaaS or hardware appliances, PAM tools manage privileged access for people (system administrators and others) and machines (systems or applications).

Where PAM falls short

Before we discuss what is needed in a modern, secure solution for cloud-native applications, let's look at why traditional PAM solutions fall short.

01

Long and complex implementation

Implementing PAM solutions can be complex and time-consuming. Integration with existing IT systems and applications can be challenging. Managing and configuring PAM solutions can require specialized skills and knowledge, which may not be readily available in all organizations. In many cases, a PAM specialist, internal or external, needs to step in.

02

Drastic changes to end-user workflows

PAM solutions often require end users to change the way they access systems and applications. Training and change management are crucial to ensure that users understand and adopt new processes.

03 Changes to Applications

Some applications may need to be modified or reconfigured to work with PAM solutions, including changing authentication mechanisms, modifying application code, or updating APIs, all of which can introduce security risks or impact to mission-critical business functionalities.

04 Lack of granularity

Many PAM solutions do not integrate directly with newer systems and applications, limiting their ability to secure access at a granular level. Instead of securing specific resources within an application, they may only be able to secure the entire application, leading to rampant over privileges.

05 No unified control plane (poor visibility)

PAM solutions require patchwork to implement, complicating the management and monitoring of access policies, suspicious activity, and compliance with security policies and regulations.

What is Privileged Access Governance (PAG)?

Privileged Access Governance (PAG) combines the existing principles of PAM – safeguarding privileged accounts and resources – but introduces a holistic and lightweight approach to easily adapt to any organization's overall objectives, risk posture and specific use cases.

How does PAG address the flaws of PAM solutions?

Direct and native integrations

PAG solutions are compatible with most modern applications and cloud providers to streamline implementation – no more disrupting or completely altering critical business infrastructure while sacrificing any access control on a granular level.

Secure API-based integrations can be set up in minutes, not months, and offer the benefit of flexibility and response to rapid changes in cloud apps and cloud providers.

User experience

PAG puts extreme focus on the user experience with the goal of adding little to no additional friction to end users, preventing creative users and decreases in efficiency.

This approach offers improved developer experience (DevX) by allowing developers to keep working with their native tools and clients of their choice, as well as leveraging chat ops such as Slack, Teams, or CLI, without having to install or adopt new tools.

A unified control plane

PAG solutions provide a single source of truth for all things privileged access: visibility into infrastructure entitlement and access architecture as well as the privileged that could be obtained by identities when needed.

9 questions to ask a potential Privileged Access Governance provider

1. What levels of access granularity is the solution capable of granting over sensitive resources?

Why it matters

Granularity is key when it comes to privilege access governance. It involves defining access permissions at a very detailed level. It considers individual databases, machines, folders, buckets, namespaces, and more that a user needs to access, ensuring that no unnecessary privileges are granted. When managing access to production, customer data or other sensitive applications , it is important to grant “just enough” access to perform the task at hand.

What to look for

Look for a solution that leverages native integrations to all your critical services, apps, and data repositories and is able to grant permissions in as high or low of granularity as is required. For example, a self-hosted and cloud-hosted PostgreSQL, MySQL, and Mongo integration can manage access to clusters, databases, collections, schemas, and more, whereas traditional PAM solutions usually stop at the cluster or database level.

The solution must be able to integrate directly with the services and the changing of the permissions at the integration level itself and speak the policy language of each one, bringing a unified privilege control plane to the admin, with workflows and audit capabilities on top.

2. How does the solution integrate with the organization's environment?

Why it matters

It is important that the solution integrates with the way the environment is set up. For example, if the organization has an integration between AWS and Okta (for example, leveraging Okta SSO), it is important that the solution grants the privileged access over that integration rather than creating a different way to get access. Similarly, if an organization leverages Terraform for some of the permission management, it is important to understand how the solution would fit on top of that.

What to look for

A solution that suits many different ways of authenticating, for example in AWS whether it is via your SAML integration, AWS Identity Center or assume role that the organization uses to give privileges to users in AWS, Apono will work in that way over the integrations you already have in place.

Similarly look for a solution that leverages TF and is fully controlled from a TF provider if your organization prefers to work with TF. These are just two small examples, but as a product the solution should understand how important it is to integrate with the way the organization works and the existing tools and processes.

3. Does the solution support time-bound, just-in-time access?

Why it matters

Just-in-Time Access Management is important because it aligns access privileges with actual needs, reduces security risks, ensures compliance, and enhances operational efficiency in a rapidly evolving digital landscape. It helps organizations maintain a robust security posture while enabling efficient and effective access to resources for authorized users.

What to look for

It's important to make sure the solution has the capabilities to dynamically grant and revoke permissions to all the critical resources and services to which it governs access. In addition, the solution should strive to offer robust and dynamic IFTTT scenarios, by leveraging context about on-call shifts, IdP groups, managers, work hours, and more to make sure Just-in-Time access is refined to the specific business use case.

4. Which cloud providers does the solution integrate with?

Why it matters

Many companies today are opting to be multi-cloud, and it's important to make sure your solution will support them all at a granular level and not just at the IAM level. Similarly, some companies have workloads running in Kubernetes or on-prem, for example with a mixture of cloud-hosted and self-hosted Kubernetes clusters and databases, and you need to make sure the solution supports both types of deployments.

What to look for

You want to make sure the solution integrates with your cloud provider and all major cloud providers such as AWS, Google Cloud Platform and Microsoft Azure, as well as Kubernetes and supporting self-hosted database integrations at the same level of granularity.

We witness many companies switching providers during the course of their lifetime for better financial incentives and you would want to make sure the solution does not need to be replaced once multiple cloud providers are in use.

5. How do end users receive the access they are granted with the solution?

Why it matters

It's important the solution is easy to use, or else it won't be adopted company-wide. To make it easy-to-use, it's imperative the solution integrates with your tech stack and doesn't require internal maintenance, for example by using home-grown solutions with automation tools, workflow builders, Slack bots and GitHub PRs. It should allow for quick, automated and simple ways to request and be granted access.

What to look for

Look for a solution that your end users are familiar with and use on a daily basis. Messaging or existing IT support / ticketing applications are often a great option, but cross reference with your organization's specific tech stack.

6. What is the overall end-user experience using the platform?

Why it matters:

It's very important that the experience from the end user side is intuitive and simple, as a large part of access governance includes a human element. To adopt a privileged access solution it must be intuitive, and it must integrate easily with the way users are already used to working. If not, it will create friction and inevitably fail to be adopted, or worse, adopted and then misused by some individuals who bypass the system and just impersonate an application token or create automations, like cron jobs.

What to look for

A policy-based access governance solution that doesn't change the way end users work and allows them to seamlessly use any client they would like to access resources and services, like cloud resources or databases.

Users should have clear visibility of their request status in their platform of choice, understand why requests were approved or rejected and time-saving mechanisms for frequently needed access.

7. What access governance automations can the solution provide?

Why it matters

Automation is an important part of any access governance solution that offers self-serve capabilities or just-in-time permissions. Not automating the revocation of permissions leads to standing privileges, resulting in a larger attack surface and potential security issues. In addition, it's also important to set up workflows with automated responses for repetitive or emergency requests.

What to look for

It's important to make sure the solution has the following:

- OnCall shift integration so that developers on-duty can request and be granted access as soon as possible if there's an incident, at any hour of the day
- Break-glass scenarios to allow different teams to gain sensitive access temporarily, for example for production maintenance, customer support, and more.
- Automation based on Cloud/Kubernetes resource tags/labels, so that new resources can be automatically included in existing access workflows

8. What access approval workflows can be defined in the solution?

Why it matters

Automatic approval workflows are huge time-savers. The user is able to seamlessly ask for and receive the access needed to do their jobs.

What to look for

- Automatic granting with full audit and reporting mechanisms in place
- Approval workflows leveraging context based approvals like on-call shifts, IdP groups and managers, and more
- Different approval flows based on resource sensitivity, i.e. data sensitivity, customer environments, or cloud account
- Approval escalation policies with multiple approvers to make sure requests are handled swiftly, and multiple approvers for very sensitive access

9. How granular is the solution inside databases or other sensitive resources?

Why it matters

To keep your resources secure, it's important to limit access to each one. Granular provisioning allows you to "check out" one book instead of the whole shelf.

What to look for

The solution must be able to integrate directly with the specific service or resource type. This allows the solution to change the permissions at the resource level itself, for example a specific collection or table in your data repository instead of the entire cluster. The solution should allow for control of specific roles and permissions of each resource type and service from one central tool, bringing a unified privilege control plane to the admin, with workflows and audit capabilities on top.

Next steps (first a little about us)

Apono is a permission management automation platform that provides simple, secure and precise just-in-time permissions across the DevOps domain. Apono is self-servable, takes just minutes to deploy, and easily integrates with your existing cloud services, Kubernetes, data repositories, and other R&D applications. With Apono, view existing permissions and easily enable dynamic, context-aware access workflows directly from Slack, Teams, or CLI.

Below you'll find how Apono answers the 9 questions above to help familiarize you with Apono and our DevOps-first solution.

1. What levels of access granularity is the solution capable of granting over sensitive resources?

Apono is a solution that leverages native integrations to all your critical services, apps, and data repositories.

Our platform is able to grant permissions in as high or as low of granularity as is required. For example, our self-hosted and cloud-hosted PostgreSQL, MySQL, and Mongo integration can manage access to clusters, databases, collections, schemas, and more.

In addition, our solution is able to integrate directly with the services and the changing of the permissions at the integration level itself and speak the policy language of each one, bringing a unified privilege control plane to the admin, with workflows and audit capabilities on top.

2. How does the solution integrate with the organization's environment?

Choose a solution that suits many different ways of authenticating, for example in AWS whether it is via your SAML integration, AWS Identity Center or assume role that the organization uses to give privileges to users in AWS, Apono will work in that way over the integrations you already have in place.

Apono also leverages Terraform and is fully controlled from a TF provider if your organization prefers to work with TF. These are just two small examples, but as a product the solution should understand how important it is to integrate with the way the organization works and the existing tools and processes.

3. Does the solution support time-bound, just-in-time access?

Apono's just-in-time solution is built with DevX in mind, allowing users to quickly ask for and receive short-lived credentials with additional context-providing fields, all in one simple form on Slack or Teams.

4. Which cloud providers does the solution integrate with?

Apono integrates with all the big cloud providers such as Kubernetes, AWS, Google Cloud Platform and Microsoft Azure.

5. How do end users receive the access they are granted with the solution?

Apono offers several different methods for receiving and granting permissions, including a web user portal, Slack, Teams, or API.

6. What is the overall end-user experience using the platform?

Apono is a policy-based access governance solution that doesn't change the way end users work and allows them to seamlessly use any client they would like to access resources and services like cloud resources or databases.

7. What access governance automations can the solution provide?

Apono offers a variety of automated workflows including the following:

- Provisioning and deprovisioning without manual IT/DevOps work
- Group membership attachment without manual Identity/IAM/IT work
- Access grant to developers on duty based on their on-call shift
- Continuous cloud discovery for all of our integrations as the environment grows or changes
- Working with CLI and Terraform for infrastructure as code based automations both for admins and end users
- Context-based approval flow allocations based on group membership, manager, shift, etc. whereas some companies deploy these complex workflows with workflow builders or ITSM ticketing systems like Jira

8. What access approval workflows can be defined in the solution?

With Apono, you can create the following workflows plus more:

- Automatic granting with full audit mechanisms in place
- Approval workflow with certain IDP group, users' manager, and other context based approvals like on-call shifts
- Approval by resource sensitivity, i.e. a tagged sensitivity or regulated data requires a higher level of approval
- Approval escalation policies with multiple approvers in cases

9. How granular does the solution provide for provisioning inside databases or other sensitive resources?

Apono is able to integrate directly with specific service or resource types. This allows users to change the permissions at the resource level itself, like a specific collection or table in your data repository instead of the entire cluster.

Our solution allows for control of specific roles and permissions of each resource type and service from one central tool, bringing a unified privilege control plane to the admin, with workflows and audit capabilities on top.

Finally...

We hope this guide provided you with a much deeper understanding of the questions you should be asking potential privileged access governance solution providers, and some of the answers to look for.

Feel free to → **Contact us** ← for help or guidance as you think through potential next steps for your organization's access management needs.