Buyers guide for security leaders looking to modernize or augment PAM

Contents

1. PAM was born on-prem

Legacy PAM was built for a world where infrastructure barely moved. Servers lived for years. Databases stayed put. Admins used shared static credentials, and the biggest risk was someone writing a root password on a sticky note.

Within those constraints, the vault-rotation-checkout model was perfectly rational. It centralized secrets, enforced basic controls and gave teams clarity about who touched what. Organizations had:

- Tight network boundaries
- Predictable infrastructure
- Human admins who needed occasional elevated access

That's why this model worked. It wasn't over-engineered. It was built for the reality of the early 2000s. And in that world, privilege was easy to define and contain.

---

2. Legacy PAM breaks down in hybrid infrastructure

Cloud and hybrid infrastructure shattered the assumptions PAM relied on. Privilege no longer lives in a handful of admin accounts on long-lived servers. Today it spans:

- Cloud VMs and databases that appear and disappear automatically
- Kubernetes clusters that churn constantly
- IAM roles, API keys and ephemeral tokens
- Code pipelines, deployment systems and automation jobs

These resources don't behave like static assets. They don't wait for a vault to update. They don't map neatly to "admin accounts." And they definitely don't like being shoehorned into workflows designed twenty years ago.

This is why legacy PAM struggles:

- Static credentials become unmanageable when resources live for minutes
- Shared accounts destroy accountability during audits
- Standing access piles up quietly in IAM
- Engineers find the workflows slow and route around them
- Every new environment demands another integration, another proxy, another maintenance cycle

At scale, the model becomes a drag on both security and operations.

3. Where this leaves modern enterprises

Almost every organization trying to stretch legacy PAM into the cloud ends up in one of a few predictable states. Understanding where you fall or what to avoid can help avoid well intentioned misadventures.

3.1 The failed vault extension

The team hooks a vault into a few cloud systems and calls it progress. But the adoption never lands. Engineers keep using SSH keys in repos and laptops. Static DB credentials stay in circulation. Cloud entitlements remain overly broad. Breakglass access is murky or nonexistent.

The PAM footprint remains where it always was … on-prem … while cloud complexity grows around it.

3.2 The partial cloud migration

Some cloud assets connect cleanly. Secrets rotate. Check-out workflows exist. But nothing feels smooth. Standing access remains everywhere. Integrations require constant upkeep. Professional services become part of the operating model.

You get coverage, but only in pockets. And every audit cycle reminds you of what still sits outside.

3.3 The homegrown bastion

To simplify access, the team builds a central bastion. It works for a while. Then scale happens. Once you're inside the bastion, access is usually broad. Shared accounts persist. Double authentication irritates developers. Auditing is limited and correlating identities to actions remains messy.

It's a clever stopgap, not a long-term strategy.

3.4 The bastion-led PAM

Tools like Teleport or StrongDM reduce some overhead and improve the session experience. They introduce developer friction with secondary authentication and sit in the critical path of access, which means they can add latency or even create downtime risks in customer environments. You're still deploying proxies everywhere.

Discovery is manual. Access maps back to shared accounts instead of real identities. Cloud entitlements and SaaS privileges sit outside the model entirely.

It's a meaningful improvement, but it doesn't close the governance gaps that keep appearing in audits.

4. The mounting blind spots

No matter which approach you're using, the same pain points show up:

- You don't have a full inventory of sensitive resources or who can access them
- Privileged accounts don't consistently tie back to IdP identities
- Static credentials stick around because nobody has time to replace them
- Standing access accumulates in cloud environments until it's a problem
- Manual approvals and ticketing can't keep pace with cloud speed

These problems aren't outliers. They're the natural outcome of forcing legacy controls onto modern systems.

Vaults and bastions absolutely have their place. But they're covering a shrinking percentage of what your engineers and workloads actually touch.

---

5. Where PAM requirements stand today

Gartner's 2024 and 2025 MQs reaffirm what "PAM" still means: vaulting, rotation, session recording, session brokering, privileged elevation. If you run on-prem systems, these capabilities are still very much a part of your reality.

That is why modernization isn't about ripping out what still serves its purpose. It's about extending PAM to adapt to the way cloud-native environments function.

That means capabilities like:

- API-driven privilege discovery
- IdP identity-native authorization
- Just-in-time access provisioning
- Short-lived or auto-rotated credentials
- Unified governance across humans, workloads and agents
- Access flows built into how engineers already operate

It's a shift from disjointed workflows and shadow authentication to seamless authorization focused controls that keep the corporate identity in tack throughout the entire process.

6. A non-negotiable: PAM for machines

The 2025 Gartner MQ adds machine access management as a required capability. That isn't a trend. It's an overdue acknowledgement of how modern environments operate.

You now need to govern:

- Workload credentials
- Machine-issued secrets
- Certificates
- SSH keys
- Machine entitlements
- Automated workflows

Machines outnumber humans in most production stacks, and they're often the least governed part of the environment. They deploy infrastructure, run CI/CD, build containers, perform automation and hold privileges long after they should.

Vaults and bastions were never designed for this surface area. In many cases, they contributed to the sprawl.

Any modernization initiative that ignores machine access will inherit the same weaknesses that enterprises are now trying to remediate.

7. How to evaluate modern PAM options

When comparing traditional approaches with identity-native ones, the differences become obvious quickly:

Capability	Vaults	Homegrown	Bastions	Identity native (P0)
Identity Mapping	Shared accounts	Shared accounts	Partial	Full IdP integration
Credential Model	Static rotated	Static	Short lived	Ephemeral, IdP issued
Infrastructure	High	Medium	Medium	Low
Coverage	On-prem	Limited cloud	Broader infra	Humans, workloads, entitlements
Experience	Complex	Manual	Streamlined	Native and automated
Fit for modern environments	Low	Low	Medium	High

Legacy systems are strong where they were designed to be strong. They just weren't built for the velocity or scale of cloud, or the explosion of nonhuman identities.

8. The P0 Security approach

P0 augments legacy PAM with continuous Zero Standing Privilege (ZSP) across users, workloads and agents. It covers the gaps legacy PAM can't reach without forcing teams to reinvent half their infrastructure.

8.1 Core design

P0 integrates directly with your IdP and your production systems. No vaults. No proxies. No bastions. Privileged resources are discovered automatically and every access path ties back to a real identity.

8.2 Key use cases

Discovery

- Build an inventory of every identity with production access
- Map owners, credentials and entitlements

Posture and governance

- Surface overprivileged identities
- Detect static credentials and excessive cloud roles

Lifecycle management

- Automate just-in-time access
- Replace static service account secrets with short-lived credentials
- Enforce expiration and rotation policies

8.3 Architectural benefits

- API-driven orchestration
- No additional infrastructure to maintain
- Developer-first workflows built around natural engineering patterns

PO turns privilege from a static perimeter control into active, continuous governance.

---

9. Business value

When modernization is done well, the benefits are obvious.

9.1 Improve security

- Remove standing privileges and static credentials
- Enforce least privilege through automation
- Reduce lateral movement paths

9.2 Reduce governance overhead

- Centralize visibility across all identity types
- Eliminate manual ticketing
- Simplify audits

9.3 Simplify operations

- No vaults or proxies to maintain
- Automatic discovery of privileges
- Higher engineering adoption

---

10. What success looks like

A modern privileged access posture has a few unmistakable characteristics:

IdP federation across all identities
Everything maps back to a real person or workload through the existing identity provider. No separate identity universe for PAM.

No standing access
Privilege is issued only when needed and only for the duration required. Nothing sits around waiting to be abused.

Short-lived credentials by default
Static keys, long-lived tokens and persistent service account secrets disappear. The system issues scoped, temporary credentials automatically.

Automated auditing and fine-grained governance
Access events map cleanly to identities. Audit prep isn't a multi-week correlation project. Review cycles are predictable instead of reactive.

High developer adoption, no user complaints
This matters more than vendors like to admit. If engineering bypasses the system because it's slow or clunky, the model fails no matter how secure it looks on paper.

Mean Time To Access (MTTA) is both measurable and minimized
This is the real barometer of whether a modern model is working. If it takes fifteen minutes, three approvals and two different authentication paths to get into production, users will route around the system. A modern privileged access layer should make access *faster* and *more predictable*, not slower.

The moment you can measure mean time to access per user, per role or per application, you can actually manage it.

Conclusion

Modernizing PAM isn't about replacing everything. It's about acknowledging where the old model still works and where it no longer fits.

Vaults and bastions remain useful for network-layer control. They just don't solve for the systems, identities and workflows that now make up most of a production environment.

An identity-native approach extends your program to meet the reality of cloud and hybrid environments while moving the organization toward continuous zero standing privilege.

The right path starts with understanding where your current investments deliver value, where they don't and where an identity-centric layer can close the gaps.

> "P0 is a game-changer. Before, we had to choose between access granularity and ease of use. Now we get both. I sleep well knowing long-standing escalated access isn't lurking in any group."
>
> **Eugene Yedvabny,**
> Senior Staff Software Engineer, Afresh

Let's change the way you manage privileged access...

Extend PAM to multi-cloud and hybrid environments with a central authorization control plane for all users, NHIs and agents.

Visit www.p0.dev for a demo or to start your free trial.

P0 SECURITY

Contact Us

Email: info@p0.dev
Web: www.p0.dev

P0 Security helps companies modernize Privileged Access Management (PAM) for multi-cloud and hybrid environments with the most agile way to ensure least-privileged, short-lived and auditable production access for users, NHIs and agents. Centralized governance, Just-Enough Privilege and Just-in-Time controls deliver secure access to production, as simply and scalable as possible.
Every identity. Every system. All the time.

P0's Access Graph and Identity DNA data layer make up the foundational architecture that powers privilege insights and access control across all identities, production resources and environments.

With P0, production access is least-privilege, short-lived and auditable by default, including the new class of AI-driven agentic workloads emerging in modern environments.