P0 SECURITY

Secure by design:

When agents enter production

Critical access controls and guardrails that must scale alongside agentic AI development

Organizations across every sector are moving from early experimentation to practical application of agent-based projects in 2026. The pressure to innovate with AI and leverage agents to achieve operational efficiencies is substantial and shared across industries. Budgets are being liberally earmarked for product development of internal and customer facing agentic workflows that will increasingly replace what once required human attention.

As these agents start to materialize, a number of common use cases are emerging:

- Customer success agents that retrieve or update account information
- Agents that support code quality or testing
- SRE and SOC style operational automations
- Financial reporting and other high stakes workflows

The business appeal is clear as these early initiatives move into boardrooms and annual planning cycles. Agents can automate repetitive work, improve information accessibility and reduce operational bottlenecks across resource intensive systems. Holding the potential to drastically decrease time to market, scale operations, minimize OpEx and more.

Yet as these initiatives move closer to production, organizations need to be especially mindful of the sensitive systems and data at play. Agents will be deployed to perform actions on behalf of human users. Think of it as a bunch of unpredictable interns with overly permissive and standing access to your crown jewels. Without proper access guardrails, the risk is substantial and the need for governance is obvious.

Agentic production access is unchecked and over-privileged

AI agents are increasingly playing a part in how modern developer teams build, automate, and scale. Whether it's using AWS Bedrock, Google Vertex, or custom-built LLM-powered systems, these agents now interact directly with cloud resources, data, and applications, often without a human in the loop.

This introduces consequential risks:

- ✓ First party agents often run with service accounts tied to broad, static roles
- ✓ Prompt engineering or model hallucinations can lead to unauthorized queries
- ✓ They can access entire databases even when the user only needs a single record
- ✓ There's no approval workflow, no supervision, and no clean audit trail

While their autonomy drives innovation, it also introduces a new kind of risk that's playing a larger role in enterprise environments. Most AI agents today operate with static credentials and overly broad IAM roles. Once deployed, they hold standing access to sensitive systems and data. Access that's rarely monitored, often unmanaged, and nearly impossible to audit due to deferred accountability.

Recent publicly noted breaches underscore why identity and access scoping must be addressed early in development, well before agentic workflows are pushed to production. Overly permissive service accounts and workload identities accessed sensitive cloud data, code repositories, and downstream systems, dramatically expanding the blast radius after compromise. These failures stem from excessive, long-lived privileges granted during system design – a pattern that becomes even more critical to correct as agentic workflows operate autonomously and at scale. The risk pattern with agents is the same: standing access and broad reach amplify risk that is only scaled when inherited by autonomous workloads that leverage those identities.

Effectively bringing agents into existing access management programs requires rethinking security teams to rethink the legacy PAM approach. Those tools were built for predictable access patterns, vaults, and stable boundaries, but agents introduce complexity and scale that makes extending coverage of these systems an operational and security nightmare. Allowing agents to automatically check out admin credentials creates unacceptable blast radius. What's needed is fine-grained, temporary access escalation with human review.

ESTABLISHING PROACTIVE CONTROL

Deferring identity work until after deployment can lead to rework or delays when risks begin to surface. Securing access for agents must be part of the project plan, not a retrofitted afterthought. As organizations prepare for 2026, four authorization requirements stand out as essential for any agent that will interact with production systems or customer data.

MCP tool access control

Agentic systems typically operate by calling internal tools (APIs or MCP-based integrations) that allow an LLM to read or modify enterprise data. In many implementations we see, these agents are backed by a single service account with broad, static permissions to simplify development. As a result, the agent can see and invoke tools or access data far beyond what the initiating user should be allowed to do.

A common example is customer-facing agents that are intended to answer questions for an individual tenant but are granted access to all customer data. Because the service account is over-permissioned, any hallucination, logic error, or prompt-level exploit can cause the agent to return or act on data outside the intended scope.

This is a familiar identity problem, over-privileged service accounts, but it becomes more dangerous in agentic systems that act autonomously. Effective control requires binding MCP tool access and data scope to identity and context, using least privilege and just-in-time permissions rather than broad, long-lived agent credentials.

Organizations adopting agents need a control point that can:

- Evaluate the requesting user or workload identity at request time
- Dynamically scope the tools and actions exposed to the agent based on that identity
- Prevent agents from discovering or invoking tools the user is not authorized to use
- Ensure agent capabilities are aligned with existing roles and policies
- Block unauthorized actions at runtime, not through post-hoc review

EARLY SIGNS YOUR PROJECT NEEDS TOOL GUARDRAILS

- When agent access needs guardrails
- Agents run with broader permissions than the users they represent
- MCP tools or internal APIs are exposed without role-based filtering
- A single agent can access data across customers or environments
- Agent actions can’t be clearly attributed to a human identity

Data access control

Some agent projects involve direct interaction with databases or internal data sources. Agents may generate queries based on natural language or call tools that support broad data access.

Without guardrails, the agent may retrieve more information than intended or leak data associated with other customers.

Early examples highlight the risk. A large payment processing company described concerns that an agent could return one customer's data to another.

Another organization noted that its agent could issue broad queries with little restriction. These situations grow more complex when the agent constructs queries autonomously.

Organizations need a way to:

Evaluate each data action before it reaches the system

Understand and inspect the structure of the query

Allow or deny the action based on the user's identity

Guardrails at the data layer prevent cross tenant exposure and ensure that autonomy does not create new leakage paths.

Just-in-Time access

Agents may require elevated rights only at certain moments. Some organizations described scheduled workloads that should have access only during the time the job runs.

Others noted batch processes that require temporary access to cloud objects. Teams at industry events asked how these Just-in-Time patterns could apply to non-human identities.

As agent capabilities expand, similar requirements will appear in user driven workflows. An agent working on a task may reach a point where it needs elevated rights.

Any escalation should be scoped, time bound and tied to an approval flow. In some designs, the agent itself may initiate the request when it determines the task requires additional permission.

Organizations adopting agents need mechanisms that:

- **Issue short lived privilege**
- **Limit access windows**
- **Support approval flows for elevated actions**

This prevents agents from running with broad, standing privilege and reduces the potential impact of unintended or unsanctioned actions.

Identity provenance and auditability

When an agent acts on behalf of a user, teams must be able to determine who initiated the action, which identity was used and what occurred inside the system. Without that clarity, teams cannot complete audits or respond to incidents.

This becomes more pressing when agents inherit user credentials. If the agent uses the human's identity without separation, distinguishing agent actions from user actions becomes impossible. The scale grows rapidly when deployments involve large numbers of agents.

Organizations need an auditable chain that:

Ties each agent action back to the responsible human

Records the steps taken during the session

Supports replay for investigations or reviews

These records help security and platform teams verify that autonomous workflows remain within approved boundaries.

QUESTIONS TO ASK EARLY IN YOUR AGENT PROJECT

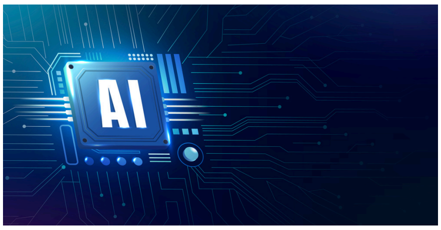
- Will the agent ever act on behalf of a human?
- Will the agent touch customer data?
- Does the agent inherit identities that have broad or standing access?
- Can we attribute every action to a specific user?

AGENTIC ACCESS TIED TO IDENTITY, SCOPE AND CONTEXT

P0's Authz Control Plane for Agents serves leaders who sit at the intersection of business enablement and security

By leveraging P0, security leaders gain governance over a rapidly expanding class of non-human identities. Platform engineering and developers can seamlessly deploy agentic productivity apps without introducing sprawl and unnecessary risk.

Right-size agentic privilege with secondary access controls, based on the human end-user that's interacting with the AI



Scale autonomous AI responsibly by enabling productivity enhancements without opening the door to ungoverned identity sprawl

Make accountability the default and audits painless with session-level replay and automated evidence trails tied to the responsible identity for simplified audit prep

With P0, secure access is not an afterthought in agentic AI development projects. It is a foundation for deploying agents in production that are secure by design.

Helpful resources:

- ↗ Technical Deep Dive: AuthZ Control Plane for Agents
- ↗ Google Vertex AI
- ↗ Access in control: AWS Bedrock

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.