P0 SECURITY

CASE FILE

Anatomy of an agentic outage

How standing privilege and unmanaged agents become a **nine second** path to complete production outage

AI agents are no longer a pilot project, they are how work gets done across the business. They are calling tools, touching production systems, reading sensitive data and acting through credentials that weren't intended for them.

That changes the potential blast radius of identity risk across a far more complex action chain.

When an agent finds a standing credential, inherits broad permissions or routes around a prompt-level guardrail, the failure is not just an AI failure. It is an access control failure.

The question is no longer "did the model make the right decision?"

The question should be: **should this agent have been able to take that action, with that authority, in that system, at that moment?**

Agentic failures result from access control failures.

In this case file, we look at how one agentic workflow moved from credential discovery to production outage in nine seconds, and where runtime access control would have changed the outcome.

CASE FILE 001

9 SECONDS from agent action to total destruction

THE INCIDENT

A long-lived API token, originally created for one specific purpose, sits forgotten in a repo file unrelated to its stated use. It carries broad permissions that lead to damaging results when an agent uses them and acts recklessy.

P0 IN THE SAME NINE SECONDS

In those same nine seconds, P0 binds the originator and agent to one blended identity that is assessed at every control point for runtime policy enforcement. The destructive command never happens.

00:00

A long-lived credential sits in the environment

A long-lived API token sits forgotten in a repo file unrelated to its purpose, with broad permissions that never expire.

Credential discovered > P0 Identity Inventory

Continuously inventories every agent, MCP server, and identity in the environment. Over-broad credentials are flagged before any agent finds them.

00:03

The agent discovers and adopts the credential

Agents are built to pursue goals. This one scanned its read scope, picked up an ungranted token, and escalated its own privilege.

Self-escalated credentials > P0 OAuth Server

Every action runs on a blended identity token binding initiator to every intermediate agent. A standalone credential carries no authority on its own.

00:05

The agent calls a destructive endpoint

No runtime authorization sits between agent and API. Prompt-layer rules are guidance, not control; the destructive call goes straight through.

Unchecked destructive calls > P0 AI Gateway

Authorizes every agent call at the platform layer: allow, deny, or require approval. The prompt has no override.

00:09

No policy decides whether the action should be allowed

The agent holds standing privilege because the token does. Nothing checks task against policy at runtime.

Standing privilege > P0 AuthZ Control Plane

Mints just-enough privilege just-in-time and revokes it when done. Every action requires an explicit policy match: that initiator, that agent, that resource.

Four phases, four missing control points. P0 enforces policy across the full agentic action chain, from identity to tool access to app-layer authorization.

The agentic access control stack

An MCP gateway addresses **one access layer**. Agentic access control **requires five**.

Discovery

Agent discovery

Discovers every agent and MCP server active across the organization

Why it matters:
You can't secure agents you can't see. This gives teams an inventory of managed and unmanaged use of agents before they are granted access to sensitive systems.

Identity

Blended identity

P0 OAuth Server captures intersecting authority between originator and agent

Why it matters:
Policy needs to know who or what initiated the action, which agent is acting, what business context applies and the resulting authority that should carry through.

Tool authorization

AI gateway

Controls which agents can call which tools

Why it matters:
This is the enforcement point for tool access. It helps determine whether an agent can call a specific MCP server, tool or connected workflow.

Resource authorization

AuthZ Control Plane

Enforces least-privileged, ephemeral access inside the target system

Why it matters:
A gateway can control tool access, but not fine-grained entitlements within the target resources. This layer enforces the actual permissions: what data, table, record, command or resource the agent can act on.

Audit

Privilege governance

Captures full provenance for every agentic action

Why it matters:
Teams need to prove what was done, on whose behalf and why access was allowed, denied or revoked. Then monitor for policy drift as the environments scale.

What a gateway alone misses

- No blended identity
- Broken attribution
- No app-layer authorization
- Standing privilege

The principle

Runtime agentic access control enable the business to confidently realize the productivity gains AI promises, without taking on the risk of unchecked access to sensitive resources and data.