

How I'd position agentic identity security, from someone who had to lock down his own AI agents

Jayden Williams · University of Maryland, Information Science, Cybersecurity minor · October 2026

Where I'm coming from

I built ContentFarm, a system where AI agents research, write, render and post short videos for brands I run. To post, the agents work inside browser sessions on real Instagram, TikTok and YouTube accounts. The first time it ran end to end, the scary part wasn't the videos. It was that the agent could do anything I could do on those accounts, all day, whether the task needed it or not. So I put limits in. I log in to every account myself and the agents only work inside those sessions, nothing goes live until two separate switches are on, and new accounts are capped at two posts a day. Then, while writing this, I found an old tool still on the agent's list that could save a password and log in. My rule said agents never log in. The code said they could. That gap is standing privilege, and it's the problem Apono exists to close.

Who buys, and what they're worried about

- **The security or IAM lead.** Agents are showing up with API keys and service accounts that never expire, and nobody can answer "what can this agent reach right now, and why?" A prompt-injected agent with standing access is an insider they never hired.
- **The platform or DevOps lead who owns the agents.** They want agents shipping work, not waiting in a ticket queue. Their fear is that security slows the rollout to a crawl or bans agents outright.

The message has to win both of them, or the deal stalls in between.

The position

For security and platform teams putting AI agents into production, Apono gives every agent only the access its current task needs, for as long as that task runs, and takes it back automatically when the task ends. Vaults and older PAM tools were built for people who log in a few times a day. Apono was built to grant access for one job and pull it back when the job is done, through Slack and the CLI where engineers already work.

ONE LINE

No agent keeps access it isn't using right now.

What I'd say, and what backs it up

MESSAGE	PROOF IN THE PRODUCT
No standing access for agents	Agent Privilege Guard applies just-in-time access to non-human identities and revokes it when the task ends
Check what the agent does against what it said it would do	Intent-Based Access Control compares an agent's declared intent with its actions at runtime
Security that doesn't slow engineers down	Access requests through Slack, Teams, Jira and the CLI, with 85+ integrations
One place for people, machines and agents	Apono inside 1Password, next to the access people already manage there

How I'd take it to market first

- **Lead with a question, not the product.** Reps open discovery with "If one of your agents got prompt-injected right now, what could it reach?" Most teams can't answer it, and that's the meeting.
- **Give the problem away free.** A free agent access check that lists every agent identity and what it can touch today. A free data-risk checker is how GuardMyData got its first \$200 MRR with no ad spend, because it shows buyers their own problem before anyone pitches the fix.
- **Start where agents already touch production:** platform teams running coding agents, and support teams whose agents can read customer records.
- **Test two stories in the first month.** Risk first ("88% of attacks exploit existing privileges", Apono's own figure) against speed first ("ship agents without a ticket queue"), and keep whichever books more meetings with each buyer.

What I don't know yet

I haven't seen Apono's win and loss notes or talked to its customers, so this is where I'd start, not where I'd finish. My first two weeks would be calls with customers and the sales team, and I'd expect them to change at least one line on this page.