Putting Boundaries Around Agents: Zscaler Supports NVIDIA OpenShell Reference Architecture for Safe AI Agents

October 21, 2026

NVIDIA draws the boundary around the agent. Zscaler extends it to everything the agent can reach.

I spend a lot of my time with CISOs, and the question they are asking me has changed. A year ago it was whether to allow AI at all. Now it's all about what they need to do to make it safe and accelerate the process of taking agents into production.

It's a fair question, but a harder one to answer. For a couple of years enterprise AI mostly answered simple user prompts and then waited for the next one. The newer agents take a complex task, work out for themselves how to get there, and keep going without checking in with anyone, which is exactly what makes them useful and also what makes them hard to govern. They've shown, over and over, that they'll find routes nobody planned for: out of a test environment, into systems they were never granted, on to actions no one signed off on. The Zscaler blog ***Zero Trust for Artificial Intelligence*** walks through the public cases of unintended agentic behavior in detail, and what struck me reading it is how consistent the pattern is. In each case the agent had more access than its task required, and nothing stood between it and the systems it touched.

I wouldn't call those agents broken. They were resourceful and doing what they were built to do, which is to find a way. When I look at these incidents I see failures of access rather than reasoning. That gap is what the work I'm writing about today is meant to close and why I am announcing that we are supporting the NVIDIA OpenShell Reference Architecture for safe AI agents which establishes enforceable boundaries for autonomous agents. Extending those controls with Zero Trust architecture, Zscaler delivers Security for AI across enterprise access, data protection, and security operations, helping organizations adopt agents while controlling access and limiting the impact of compromised behavior.

The user we didn't plan for

When Zscaler started, more than fifteen years ago, the idea that nobody should sit on your network was a tough sell to all but a handful of pioneers. The argument was simple enough. A trusted exchange should broker each connection between a user and the internet or application they're trying and entitled to reach, verify it, inspect it inline, and allow nothing more. That same Zero Trust thinking runs through how we secure the enterprise at every level, and it's the foundation we built Zscaler Security for AI on when it became clear that agents were going to be the biggest shift CISOs and industry have had to absorb since the internet itself.

For access purposes, an agent is another user. However, the differences between an agent and a human user are dramatic. An agent can be created in seconds, it keeps working after the person who started it has logged off, and it can multiply. And too often we blindly hand it the full credentials of whoever invoked it, which is the part that worries me most.

So there's one question I'd put to any agent deployment: for each connection this agent makes, who decided it was allowed? If the answer is the agent, or the prompt, or nobody in particular, that harkens back to a complex access problem, whatever model is running underneath.

Zero Trust for Artificial Intelligence sets out nine zero trust principles for answering that question, from inventory and flow control through to access governance. I encourage you to check it out for the full overview. To understand the significance of our announcement with NVIDIA, let's explore what happens when two of those principles get a home in runtime and silicon.

NVIDIA draws the boundary. Zscaler extends it.

None of this is new to computing. Operating systems don't let user-space code police itself; enforcement sits below the layer it protects, and we stopped arguing about that decades ago. NVIDIA OpenShell Reference Architecture for safe AI agents brings the same discipline. NVIDIA's ownAI safety and security team says it well in ***Where Security Fits in an AI Agent Stack*****:** the harness guides what an agent tries, and the infrastructure determines what it can do.

The NVIDIA OpenShell Reference Architecture for safe AI agents combines OpenShell secure runtime with NVIDIA Vera Rubin for inference, NVIDIA Vera CPUs for agent harnesses and tool use, and NVIDIA BlueField-4 DPUs running NVIDIA Sentry for continuous, out-of-band monitoring and in-silicon enforcement.

NVIDIA OpenShell is an open source runtime that contains the agent and governs what it can see, do, and reach, which in zero trust terms is flow control built into the place where the agent actually runs. NVIDIA OpenShell governs agent execution and access through policy enforcement outside the agent's process, including kernel-level sandbox controls. The second, NVIDIA Sentry, runs on NVIDIA BlueField-4 and monitors agent activity and enforces policy independently in silicon. Think of it as continuous evaluation and containment from somewhere the agent can't touch, providing another layer of protection for high-stakes environments.

Today, alongside NVIDIA's announcements at NVIDIA GTC Berlin, Zscaler is joining the ecosystem building on that platform, and we're working with NVIDIA to bring Zscaler Security for AI based on the OpenShell Reference Architecture.

Agent workflows span runtime environments, external APIs, internal data stores, and cloud identities. Building on NVIDIA's runtime and hardware enforcement layers, Zscaler brings Zero Trust enterprise-wide access controls, data protection, and security context to those interactions.

An agent that can reach nothing isn't much use to anyone, so to do real work it has to call external APIs, read from internal document stores, and operate under cloud identities. Much of that happens a long way from the cluster where the agent runs, often on infrastructure that has no dedicated hardware enforcement at all. NVIDIA OpenShell governs agent execution: which resources and data agents can reach, and where its inference is routed. Zscaler picks up at the moment the agent reaches beyond the runtime. Through the Zero Trust Exchange, governing each call by identity and

policy, with inline inspection, data protection, and security context across external APIs and models, MCP servers and tools, internal applications, data stores, and cloud/SaaS services. And because agents will not all run inside a governed runtime, Zsacler extends the same visibility and control to agents operating on endpoints and in cloud workloads, so security teams see and govern the full population of agents in the enterprise, not only the sanctioned ones. Together, this gives enterprises containment at the runtime and consistent control everywhere the agent runs and across everything it touches.

Containing Unbound Agents

Take an agent running in an OpenShell runtime that's been asked to reconcile customer records. Partway through, one of the documents it retrieves turns out to contain hidden instructions telling it to send those records to an outside address.

The agent's identity is the same, its credentials are valid, and the task it was given is legitimate. The only new thing is the connection it's about to make. With inline inspection in the path, the retrieved content can be examined as it arrives, so the injected instruction is flagged before the agent ever acts on it. And if the agent goes ahead and tries the transfer anyway, the request is evaluated against policy: this agent, on this task, has no reason to be sending customer data to that destination. The data is stopped in transit and the attempt is recorded. That’s the outcome.

AI Guard and OpenShell

OpenShell governs how an agent executes and where inference goes. What Zscaler AI Guard is designed to add at that crossing point is the organization's own AI policy, applied inline to prompts and responses, looking for prompt injection, jailbreak attempts, exposed secrets, sensitive data on its way out, and more. We've done a version of this with NVIDIA already. Earlier this year we released an AI Guard integration for NVIDIA NeMo Guardrails, and our work on OpenShell takes the same idea one level deeper, from the application into the runtime.

AI Guard doesn't work alone. Zscaler AI Gateway, a product within the Zero Trust Exchange, brokers agent connections and scopes access to the task, and it's what extends the exchange we built for users so that it's ready for agents too. AI Access Graph, for its part, shows where agents are holding more privilege than they need.

Where Sentry comes in

Inline control tells you what an agent may do. It doesn't, on its own, help a security team move quickly when something goes wrong, especially if an agent manages to bypass software boundaries entirely. This is where NVIDIA Sentry comes in. NVIDIA Sentry, built on NVIDIA DOCA and running on NVIDIA BlueField-4 DPUs, continuously monitors agent behavior and enforces security policies from a separate trust domain outside the host. When NVIDIA Sentry detects anomalous behavior or policy violations, it can enforce policy in real time and quarantine agents within milliseconds, while providing trusted telemetry to enterprise security platforms for investigation and response.

But raw telemetry is valuable only if it reaches the people and systems that can act on it. By the time a SOC analyst manually traces rogue agent behavior back to a specific container, agent, and identity, the threat has already moved.

Zscaler Exposure Management ingests this out-of-band telemetry from NVIDIA Sentry and NVIDIA DOCA Argus directly into the Zscaler Data Fabric for Security. This automatically correlates infrastructure signals with rich, real-world context, including the specific GPU node, container pod, application owner, and business criticality. By mapping these events against exposures, exploitability, identity, and active CVE threats, the system instantly prioritizes risk. For example, an exposure on a production cluster hosting proprietary model weights is automatically given a higher criticality than an identical exposure in a development sandbox.

Zscaler Deception adds a proactive layer to this defense, deploying synthetic API keys, lure vector databases, and decoy agent endpoints across development pipelines. When an agent begins probing unauthorized pathways, these decoys trigger deterministic alerts that enable containment before any real asset is ever reached.

With this new silicon-to-cloud visibility, our Agentic SecOps solution will soon be able to correlate NVIDIA DPU telemetry with zero trust, endpoint, identity, and cloud signals to determine the complete attack chain and its associated risk. It will then be able to execute coordinated containment across the entire attack chain: isolating endpoints, terminating rogue processes, revoking identity tokens, and updating Zero Trust Exchange policies to sever lateral access, without waiting on a manual handoff.

How detection inside the cluster will become prevention

Our team and NVIDIA's are working closely together to make sure that what one environment learns can protect the others: enterprise context informs what the silicon enforces, and what the silicon sees adds to enterprise context. Soon, when NVIDIA Sentry identifies an active exploit inside an AI cluster, our ThreatLabz researchers will analyze the behavioral signatures and forensic indicators. The resulting findings will automatically generate preventative inspection rules that propagate across the global Zero Trust Exchange, often triggering the creation of new detectors in Agentic SecOps to extend that protection to workstations, cloud workloads, and employees across the organization. And these controls will run in both directions. When Zscaler Agentic SecOps identifies a distributed campaign at the network or identity layer, it will soon be able to push verified behavioral detection policies directly back into the NVIDIA BlueField in-silicon policy engine, programming hardware-accelerated filters into the DPU for line-rate packet drops and workload isolation.

Built in the open

OpenShell is designed to be open source, and a meaningful share of its features now comes from teams outside NVIDIA, people who found concerns in production and sent fixes back. For infrastructure that many organizations are going to rely on, I think that's the right way to build, and we plan to be an active participant in it. It's the same reasoning that made Zscaler an inaugural partner in the Open Secure AI Alliance and a founding member of the Blueprint Alliance.

What I would tell a CISO today

Nobody I talk to is asking how to block agents. They're asking how to turn them safely. If I had to pick three places of advice for a CISO of where to start, they are these:

1. **Ask who decides.** For every connection an agent makes, get an answer that isn't the agent.
2. **Inspect in the path.** Logs will tell you what happened. Inline inspection changes what happens, which matters when an agent can make a hundred calls before anyone looks.
3. **Choose controls that travel.** Your environment is going to be open, closed, hybrid, and multivendor all at once (it probably already is), and most of it won't sit behind dedicated hardware. Prioritize controls that work across models, runtimes, and vendors.
4. **Connect the loop.** Feed deep cluster telemetry directly into your SecOps workflows. By combining silicon-level visibility with zero trust policies, you create a continuous feedback loop that turns local detection into global prevention.

The foundation for autonomous AI is being laid in the moment we're in right now, and choices like these tend to stick around. I'm glad Zscaler is doing this work arm in arm with NVIDIA, and that we're doing it in the open.

To go deeper, check out the recent blog "*Zero Trust for Artificial Intelligence*" or visit our Zscaler Security for AI and Agentic SecOps web pages.

Safe Harbor

Forward-looking statements: This blog contains forward-looking statements that are based on our management's beliefs and assumptions and on information currently available to our management. These include statements about planned integrations, product capabilities, and our work with NVIDIA. Forward-looking statements are subject to risks, uncertainties, and changes in circumstances, and actual results may differ materially. Any unreleased services, features, or integrations referenced here are not currently available and may not be delivered on time or at all. Customers should make purchasing decisions based on services and features that are currently available. Zscaler assumes no obligation to update any forward-looking statements made in this blog.