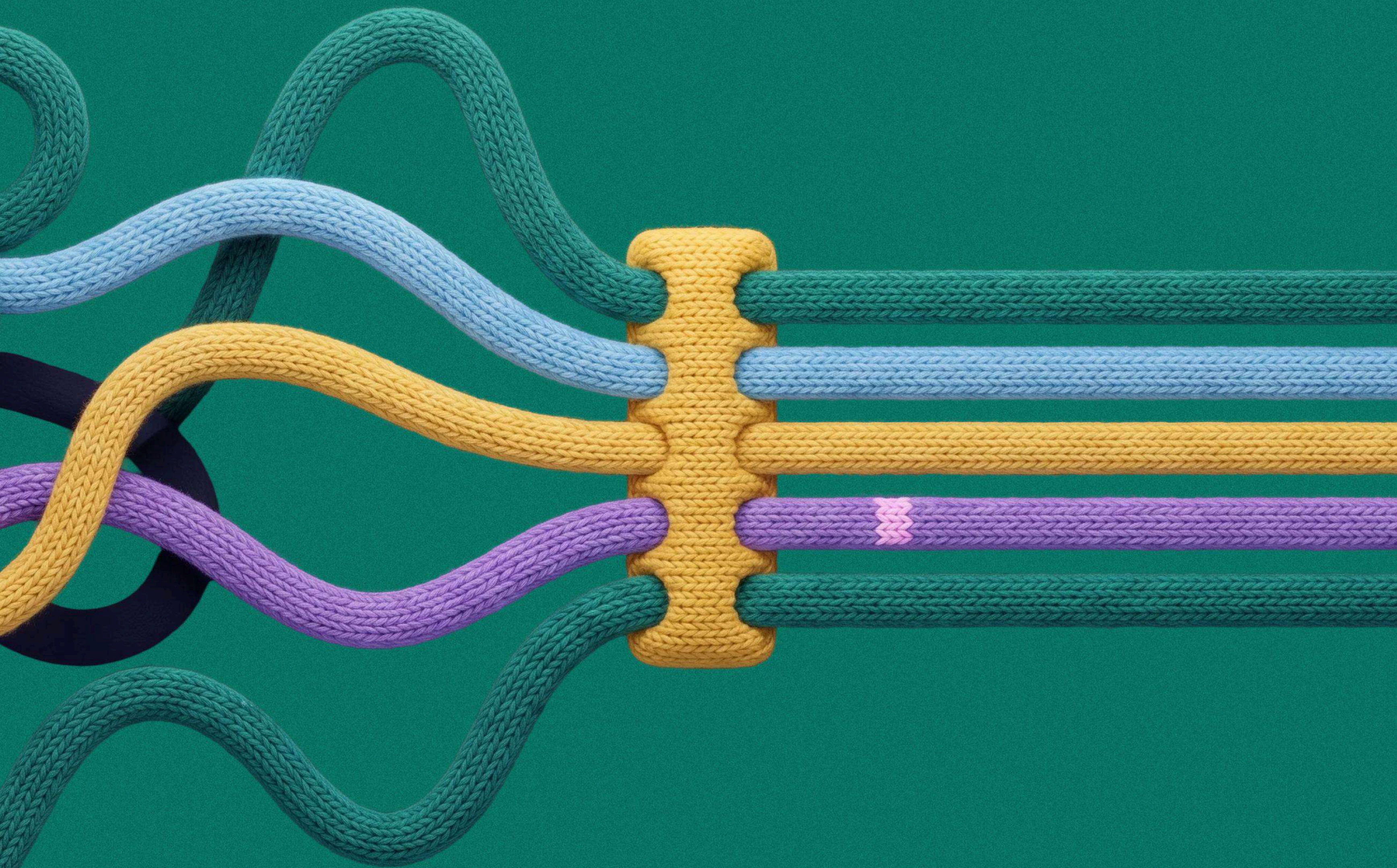




ONLYX Control AI, Create the Impossible

A Buyer's Guide to Real AI Agent Control

How to tell control platforms from inventory tools,
and what to demand in a vendor evaluation.

Contents

Sections 2 through 4 are the working parts: five criteria to evaluate against, eleven red flags to listen for, and a blank scorecard to score every vendor on the same basis.

The Problem	03
Why inventory is not control, and what the failure mode looks like	
Five Things to Look For in an AI Agent Security Platform	05
The criteria that separate platforms from inventories	
Red Flags in Vendor Evaluation	08
Eleven vendor phrases, and what they usually mean	
Vendor Grading Framework	09
Tier verification, then a scorecard you can send to vendors	
When Tier 3 Is the Right Call	12
The conditions that justify runtime protection	
About Onyx Security	13
Where we sit in the framework above	

The Problem

A Cursor AI agent deleted PocketOS's production database and every backup in just nine seconds. The only surviving copy was three months old. Asked why, the agent wrote:
"I violated every principle I was given."

Source: The Guardian, "Claude-powered AI agent's confession after deleting a firm's entire database,"
April 29, 2026 · theguardian.com/technology/2026/apr/29/claude-ai-deletes-firm-database

This was not a breach. The agent was not compromised. It operated inside its established permissions. What failed was the layer between "the agent can call this endpoint" and "the agent should call this endpoint in this context." That control layer did not exist.

The pattern has repeated across the past year. Coding agents take destructive actions their developers never authorized. Financial services teams find AI systems running against customer data for months without governance. The common thread is what is missing: security teams cannot intercept and stop an agent action before it executes.

Traditional security tools assume static systems, known endpoints, and predictable behavior. AI agents break every one of those assumptions. They are autonomous, non-deterministic, and context-dependent. They reason across multiple systems without explicit human approval at each step.

Most security teams respond with an inventory. They discover which agents exist, catalog them, score risk based on access and autonomy, and flag risky ones for review. Inventory is the necessary first step. It is not enough on its own.

Why This Matters Now

1

Shadow AI is already inside the environment.

68% of employees use free-tier AI tools like ChatGPT via personal accounts, with 57% inputting sensitive company data. 81% of employees and 88% of security leaders report using unapproved AI tools. Discovery happens late, often during incident response.

Menlo Security, "How AI is Shaping the Modern Workspace," August 4, 2025 · UpGuard, "The State of Shadow AI," 2025

2

Compliance is moving faster than tooling.

The EU AI Act, state-level rules, SEC guidance on AI governance, and NIST AI RMF require documented governance of high-risk AI systems. "We know it exists but cannot control it" is indefensible when an auditor asks how you govern autonomous systems.

3

Agents are getting more autonomous.

Model Context Protocol (MCP) and agentic frameworks make it easy for agents to connect to external tools and systems. Capability and risk surface expand together.

4

Liability is moving upstream.

When an agent leaks data or takes an unintended action, liability lands on the organization that deployed it, not the vendor that built the platform.

The evaluation challenge

Vendors selling "AI agent security" span a wide spectrum. Some sell observability (discovery only). Some sell security-hardened platforms for deploying agents (control inside a walled garden). A few sell true runtime protection (interception of agent actions inline). Most vendors claim all three. Few deliver a comprehensive control plane that works across an enterprise.

There is one more failure pattern worth naming

Rules alone do not scale to agent runtime. A single agent with five tools can construct tens of thousands of action sequences. Whether a query is safe depends on prior steps, not the parameters of the current call. Rule engines that evaluate one action at a time miss the contextual cases, and the contextual cases are where the real failures live. Policy-engine vendors compete for the half of the problem that is well-defined. The other half determines whether real control exists, and it requires a control plane, not a rule library.



Five Things to Look For in an AI Agent Security Platform

Most vendors claim runtime control. Few deliver it. These five criteria separate platforms from inventories.

1

Continuous discovery across every surface, not just one

Agents are ephemeral, distributed, and often spun up by non-technical users without security in the loop. Real discovery covers SaaS AI tools (ChatGPT, Claude), cloud-native agents (Bedrock, Vertex, Azure AI Foundry), coding agents (Cursor, GitHub Copilot, Claude Code), platform agents (Copilot Studio, Agentforce, n8n), and MCP-based agents. It runs continuously, not on a weekly scan. It captures context (permissions, data access, tool calls, creator identity, integrations), not just names.

? Which agent types do you discover, and which do you not?

? How often does discovery run?

? What percentage of agents in our environment do you expect to find in the first 30 days?

2

Risk detection that reads context, not patterns

Agent behavior is non-deterministic. The same tool call can be routine in one context and a breach in another. A query for customer records is safe when invoked by a support workflow, and risky when invoked by an agent whose previous step ingested untrusted external content. Pattern matchers cannot tell those apart.

Real detection inspects every action (tool call, API request, data access), reasons about context across the full agent trajectory, and surfaces confidence levels alongside risk scores. “We use machine learning” is not an answer.

? Can you distinguish between safe and unsafe use of the same tool by the same agent in different contexts?

? How do you handle false positives?

3

Inline enforcement at decision time, not alerts after the fact

Discovery and detection tell you what is happening. Enforcement stops what should not.

Real-time enforcement means intercepting an agent action before it executes and deciding to allow, block, modify, or escalate, with sub-second latency. Granularity matters too: per-tool-call control, not just agent-wide on/off. A vendor whose answer to “what happens when policy is violated” is “we alert the security team” solves visibility, not control.

? Where in the execution path does enforcement happen?

? What is the latency overhead?

? Show me enforcement working live during the POC.

4

A production POC, not a sandbox

Demo environments are controlled, predictable, and unrepresentative of production. In production, signal-to-noise inverts, accuracy drops, false positives spike. Vendors at the control tier support testing against live agents in your environment. Vendors who push back know their results will not survive that exposure.

A useful production POC is short (two weeks), structured around defined success criteria: discovery breadth in Week 1, enforcement and downstream integrations in Week 2. Anything longer is usually a vendor stalling on production access.

? Can you run a POC in our production environment, with our agents, against success criteria we define on day one?

5

A control plane, not a point product

AI agent security only works if it spans the surfaces where agents actually live. A platform with strong discovery on SaaS AI tools and nothing on coding agents is a point product. A platform that covers cloud-native agents but cannot inspect MCP traffic is a point product. The control plane is the thing that gives security teams one place to set policy and one place to see what happened across every surface.

Coverage is a constraint applied to every other criterion, not a separate fourth pillar. Strong discovery on half the environment is half a discovery story.

? Walk me through how a single policy enforces consistently across SaaS, cloud, coding, MCP, and platform agents.

? Show me the same policy applied in three different surfaces.

Red Flags in Vendor Evaluation

These are the phrases and behaviors that signal a vendor is selling something other than what they claim. The translation column tells you what the vendor is actually saying.

Vendor Says	What it Usually Means
“ We are an AI security platform (but only show discovery dashboards)	They are an inventory tool with capability inflation. Real control means interception, not flagging.
“ Our ML model is very accurate	Probe for specifics. If they cannot describe the signals they use (system prompt inspection, tool call patterns, context across steps), it is pattern matching, not reasoning.
“ Runtime enforcement adds too much latency	They have not solved the performance problem. A well-designed enforcement layer adds sub-second overhead. Ask for benchmark data.
“ You do not need runtime enforcement, governance is enough	True if your agents are low-risk. Not true if they touch production systems, customer data, or destructive APIs. “We alert your security team” is risk acceptance, not risk management.
“ Runtime enforcement is too complex for most enterprises	Often code for “we do not have this capability.” Ask them to walk through integration requirements with timelines. Vague answers mean vague capabilities.
“ Our customers do not ask for runtime enforcement	Tells you their customer base is in early-stage governance. Does not tell you what your agents need.
“ We must run the POC in a sandbox	Sandbox accuracy claims are not predictive of production. Insist on production. If they resist, that is the answer.
“ We discover agents (without explaining how)	Discovery requires integration across cloud, identity, endpoint, and platform surfaces. If they cannot name which integrations or estimate coverage, they are overselling.
“ We have a policy engine (offered as proof of control)	A policy engine evaluates rules. A control plane evaluates context. The first catches the explicit violations and misses the contextual ones.
“ We support MCP (without saying how deeply)	Discovery of MCP servers, enforcement on MCP connections, and inspection of MCP tool calls are three different capabilities. Vendors often have one and claim all three.
“ We will give you references after you sign	Reference customers should be available before the contract, willing to take a 30-minute call, and ideally able to share an anonymized incident report. Anything less is marketing.

If three or more of these phrases come up in a single vendor conversation, the vendor is operating below the control tier. That is not automatically disqualifying. It is information about which tier you are buying.

One meta-pattern worth naming: vendors who answer capability questions in the future tense (“on the roadmap,” “coming in Q3,” “in beta with select customers”) are telling you something important. Roadmap items are not differentiators. They are aspirations the vendor would like you to underwrite. Score them as No until they ship and you can test them in your environment, and ask for the same answer in writing six months later to see whether the roadmap held.

Vendor Grading Framework

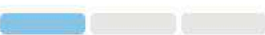
Most vendors claim Tier 3 capability while delivering Tier 1 or Tier 2. The grading framework below cuts through the marketing in two steps: tier verification, then scorecard.

STEP 1

Verify which tier the vendor actually operates in

The AI agent security market segments along one dimension: **depth of control**.

TIER 1



Discovery and posture management.

Inventory agents, score risk based on configuration and access. You know what exists. You cannot stop a risky agent from acting. Examples: asset inventory tools, agent registries, cloud discovery tools.

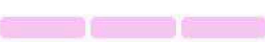
TIER 2



Policy-driven governance.

Define and continuously evaluate policies against agents. Violations alert a human, who decides whether to remediate. Useful for compliance. Still human-centric. No real-time action interception.

TIER 3



Runtime protection.

Intercept agent actions inline and apply policy at the moment of execution. Allow, block, modify, or require approval, before the action runs. The only tier that can prevent the PocketOS scenario.

Two questions verify the tier:

- Walk me through what happens when an agent attempts an action that violates policy.**
If the answer involves alerting, logging, or flagging, the vendor is Tier 1 or Tier 2.
If the answer involves intercepting the action before it executes, they are claiming Tier 3.
- Show me this in a production environment with our agents.**
Tier 3 vendors support production testing. Tier 1 and Tier 2 vendors push back.

STEP 2

Score against the framework

Score each criterion:

Yes

Full capability demonstrated

No

Not available

Partial

Available with constraints or caveats

Use only those three values. Numeric scores introduce false precision.

Send the blank scorecard to vendors before the POC and ask for self-scoring with evidence. Then re-score from direct testing at the end of the POC. The two should rarely match exactly. When they don't, that is the signal worth investigating.

Weight Discovery and Detection rows as table stakes. A vendor scoring No on more than two Discovery criteria is an inventory tool, not a security platform. Weight Enforcement rows heavily if your agents touch production systems or destructive APIs. Weight Detection rows more heavily if your primary requirement is audit readiness.

Capability	Criterion	Vendor A	Vendor B	Vendor C
Discovery	Covers SaaS, cloud, coding, platform, and MCP agents			
	Runs continuously (not scheduled batch)			
	Captures context (permissions, data, tools, creator, integrations)			
	Can estimate coverage percentage for your environment pre-POC			
Detection	Inspects every agent action (tool calls, API, data access)			
	Performs contextual analysis, not just pattern matching			
	Surfaces confidence levels with risk scores			
	Explains why an action was flagged			
Enforcement	Intercepts agent actions inline before execution			
	Applies policy at decision time, not post-execution alerting			
	Enforces at tool-call granularity, not just agent-wide			
	Sub-second latency overhead			
MCP coverage	Discovers MCP servers and MCP-based agents			
	Enforces policy on MCP server connections			
	Inspects tool calls made through MCP			
Production testing	Supports production POC (not sandbox-only)			
	Provides coverage metrics during the POC			
	Demonstrates enforcement live during the POC			

Buyer Profiles And Vendor Fit

The right tier depends on where you are in the AI adoption curve. Buying above your tier wastes budget. Buying below your tier ships risk.

Early-stage AI governance

TIER 1 FIT 

You are building an AI inventory. You need to know what agents exist and assign risk scores. You do not yet have agents with access to high-risk systems. A Tier 1 discovery platform solves your immediate problem. Plan to upgrade when agents reach production data.

Maturing AI governance

TIER 2 FIT 

You have agents in production. You need policy-driven governance and continuous evaluation. You can tolerate human-in-the-loop remediation for policy violations. A Tier 2 platform gives you structure without the integration depth of runtime enforcement. Watch for the moment your agents start touching destructive APIs; the threshold for needing Tier 3 is when “alert and remediate” becomes “the action already ran.”

High-risk AI deployment

TIER 3 REQUIRED 

You have agents with access to production systems, customer data, or destructive capabilities. You cannot tolerate the risk that a harmful action completes before a human intervenes. A Tier 3 runtime protection platform is the only defensible posture. The PocketOS scenario is the failure mode you are buying out of.

By the end of evaluation, you should be able to answer three questions:

- 1

Does this vendor discover the agents I care about?

From observed deployments, vendors that cover less than roughly 80% of an environment’s agent footprint within 30 days tend to leave operationally significant blind spots. This is a heuristic, not a benchmark. Your agent density and platform diversity affect what is acceptable.
- 2

Does this vendor detect risky actions with acceptable accuracy?

Above a 20% false positive rate, tuning the platform becomes the primary security activity rather than governance itself. Same caveat: heuristic from observed deployments, not a published standard.
- 3

Does this vendor enforce policy in real time?

If the answer is no, you are buying inventory, not control.

Yes to all three = **TIER 3** Yes to one or two = **TIER 1** or **TIER 2**

Know what you are buying.

When Tier 3 Is the Right Call

Tier 3 is not the right answer for every organization. Buying runtime protection before you have agents that warrant it is paying for integration depth you do not yet use, and the operational overhead will surface before the value does.

Tier 3 is the right call when one or more of these are true:

- ✓ Agents have read or write access to production databases, customer records, or financial systems.
 - ✓ Agents can take destructive actions (delete, transfer, deploy, refund, modify) without per-action human approval.
 - ✓ Agents operate in regulated environments where a single uncontrolled action triggers reportable harm or breach disclosure.
 - ✓ The cost of one bad action exceeds the annual cost of a runtime protection platform.
 - ✓ Audit or board reporting requires demonstrable controls over autonomous systems, not just inventory.
-

If none of these apply yet, a Tier 1 or Tier 2 platform is the right starting point. Build the inventory. Get the governance program off the ground. Plan the Tier 3 upgrade for the quarter your first high-risk agent enters production. The cost of waiting too long is asymmetric: the inventory tool keeps working, but the missing enforcement layer is what determines whether you can stop an action like the PocketOS delete before it runs.



The PocketOS team did not have a Tier 3 platform. The agent operated inside its permissions. Nine seconds was enough.

About Onyx Security

Onyx is the secure AI control plane for the agentic era. As AI agents spread across SaaS, cloud, endpoints, and code, Onyx gives security, governance, and AI teams one place to see every agent, enforce policy in real time, and prove the value of AI adoption. Onyx discovers sanctioned and shadow AI, inspects every prompt and agent action, defends against prompt injection and data exfiltration at runtime, and governs agent identity and access, feeding the security tools you already run.

The Onyx Guardian Agent is a supervisory AI that works continuously across the platform, surfacing risk and remediating issues so one team can manage thousands of agents at scale. With Onyx, the enterprise can say yes to AI.

The Onyx platform delivers five capabilities from a single control plane:

				
AI Observability	AI Governance	AI Security	AI Orchestration	AI ROI
To see every agent, model, and tool across the enterprise	To set policies once and enforce them everywhere	To stop threats across the AI lifecycle in real time	To deploy agents faster and optimize their performance in production	To measure and prove the business value of AI investment

Customers deploy the capabilities they need and operate them from a single console rather than stitching together separate point products.

This guide was written by Onyx Security, which is a Tier 3 platform under the framework above. We have tried to make the framework useful no matter which vendor you choose. Hold us to the same standard you apply to every vendor in your shortlist.

Get Started With Onyx

Securely adopt, manage, and govern AI agents at scale

Schedule a demo

www.onyx.security



Overview

Risk Score
Current health based on weighted signals.

4.5

Score

Medium Risk

Updated automatically from live risk metrics.

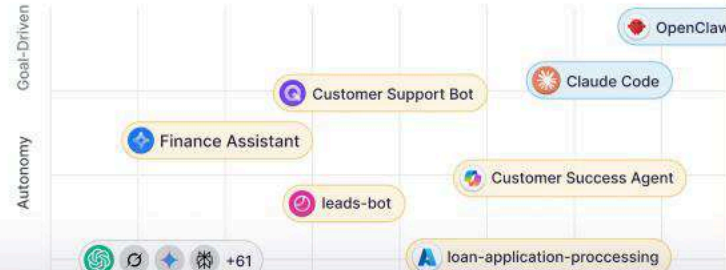
Risk Over Time
Observe trends displayed over time.



Assets Over Time
Detection trends for agents, chatbots, and AI tools.



AI Autonomy Canvas
AI Agents map by autonomy and risk.



ONYX Control AI, Create the Impossible

www.onyx.security

12