Geordie

WHITEPAPER

The CISO's guide to agentic security and operations

How security leaders run an agent estate that is already doing the business's work, from the first inventory to a program that scales.

Second Edition: Nov 2026

IN BRIEF

Running security for agents that are already at work

Most security leaders no longer need convincing that agents are running in their organization. A first inventory has usually settled that, and agents have since moved out of pilots in engineering and operations and into work the business relies on every day. The question we hear now is a practical one: how do you run security for an agent estate that grows faster than any review cycle, that spans endpoint, cloud, code and the browser, and whose ownership is still being argued over between security, IT, platform engineering and a new generation of AI leads?

This guide answers it from the CISO's seat. It draws on our work with organizations that are running agents at real scale, on conversations with more than 60 of them about who owns agent security and operations, and on what we have found inside their estates. Where we quote security and technology leaders we have kept them anonymous, because the pattern matters more than the name, and because most of them are working through the same questions you are.

WHAT THE DEPLOYMENTS SHOW

- **The inventory is still the first thing to be wrong.** Owkin found 327% more agents than its security team expected in a single proof of concept, and estates keep growing after that first count is taken.
- **Ownership is inherited rather than designed.** Security most often holds agent risk by default, and four of the recurring tasks an agent estate demands still have no settled owner in most organizations.
- **Risk and value are read from the same record.** The session trace that shows a coding agent pulling in a malicious package is the same evidence that shows 19% of one firm's AI spend running through personal accounts.
- **Controls that stop the work get routed around.** The programs that scale roll enforcement out in rings, monitoring first and guiding agents before anything is blocked.

60+
Organizations in our research

1 in 2
Agents to employees, as one large enterprise estimates its own estate

40%
Of enterprise applications forecast to feature task-specific AI agents by the end of 2026, up from under 5% in 2025 (Gartner)[1]

WHAT THIS GUIDE COVERS

01 The estate you are already running
02 Why agents break the security playbook
03 What we are finding in agent estates
04 Seeing every agent, wherever it runs
05 Controlling what agents can reach
06 Enforcement that keeps the work running
07 Risk and value from one record
08 The ownership problem
09 What to do in the next two quarters

Agents are already doing the work, and arriving by four routes

A year ago, the honest description of most enterprise agent programs was a handful of pilots in low-risk, high-friction corners of the business such as meeting summaries, note-taking and internal knowledge search. Those pilots have since grown up. Agents now write and review production code, triage tickets, reconcile financial data and draft customer communications, and they do it with real credentials against real systems of record. The board conversation has moved with them, from whether to adopt agents to how quickly the business can give them more responsibility, with one eye on Gartner's forecast that over 40% of agentic AI projects will be canceled by the end of 2027 for reasons that include inadequate risk controls.[2]

That shift changes what a security leader is being asked to do. The early work of defining an agent and finding the first few is mostly done, and what remains is the harder, continuous work of running security for an estate that changes every week. It helps to be precise about the unit being counted, because every later decision reads from the inventory.

The definition we work from. An agent is a large language model connected to at least one tool, whether an API, an MCP server, a skill, an extension, a plugin or a SaaS connector. Chat with no tool connections is not counted. The definition is deliberately minimal because it survives contact with a real estate: it puts a copilot and a multi-agent system in the same inventory, and it does not depend on what a vendor chose to call its product.

Platforms the business bought

Copilot Studio, Agentforce, Bedrock and the agent builders inside SaaS tools, sanctioned at the platform level but configured team by team.

Developers' coding agents

Claude Code, Cursor, Copilot and Codex, running on laptops with a developer's own access and a growing set of MCP servers and skills.

Workflows built by business teams

Agents assembled by finance, marketing or operations in a no-code tool, which never pass through an engineering pipeline.

Tools employees adopted themselves

Browser agents and personal accounts brought into work for productivity or preference, often outside sanctioned tooling.

> “We stop debating intent and look at capability and autonomy. The moment an LLM can observe state, decide a next action, and execute that action, especially across system boundaries, it's an agent in our threat model.
>
> **LEO CUNNINGHAM** · CISO, Owkin

Traditional software executes rules, while agents make decisions

Automation and traditional software operate on predetermined logic, a sequence of if-then conditions that produces the same outcome every time it sees the same input. A SOAR playbook for phishing triage is a good example: it extracts indicators from an email, checks them against threat intelligence and follows a fixed path of rules, which makes it easy to audit but limits it to the scenarios someone scripted in advance.

An agent given the same job behaves quite differently. It reads the whole email, interprets intent, reasons about context and decides for itself which actions to take next, whether that means querying an external source, reviewing past incidents or drafting an internal update. Its behavior adapts to what it observes, which helps in unfamiliar situations and also introduces decisions that are inconsistent from one run to the next, reasoning that is hard to reconstruct afterward, and the possibility of actions or data access nobody approved.

QUESTION	SCRIPTED AUTOMATION	AGENT
What decides the next step?	A rule written in advance	The model, weighing instructions against everything in its context
Same input, same output?	Yes, every time	Not reliably, because the path can change between runs
What does an audit read?	The playbook and its logs	The sequence of decisions, tool calls and data the agent actually touched
How does it fail?	Loudly, at the edge of the script	Quietly, by acting logically on a wrong inference
What do you secure?	The code and its access	The agent as a system: configuration, identity, tools, data and behavior

Figure 1. The same task run by a SOAR playbook and by an agent, and what changes for the security team.

Agents are systems, not surfaces

The second reason the old playbook struggles is architectural. An agent's configuration lives in one place, its tools in another, its data access in a third, and its behavior spans all of them, so a control built around a single surface or a single type of traffic sees only the slice of the agent that crosses its boundary. Configuration tells you what an agent can do, and only behavior tells you what it actually did, which is why the programs that work treat the agent's own record as the primary evidence.

> “ Once you understand that agents operate differently from traditional automation, you stop treating them like sophisticated workflows. Playbooks can't secure systems that choose their next action based on context. Instead, you focus on which decisions you're delegating, how tightly they're bounded, and how you'll monitor behavior over time.
>
> ANTONIO BOVOSO · Veteran cybersecurity leader

The risks are familiar in kind and new in shape

Much of agent risk extends what security teams already know. Delegated credentials can be hijacked, prompt injection can arrive through an email or a document payload, and data can leave through an unsafe tool call or a corrupted memory entry. The controls that answer these are the familiar ones of validating identity, managing access and protecting data, and OWASP's Top 10 for LLM applications names prompt injection and excessive agency among the primary risks for exactly this reason.[3] What has changed is the vector: a single poisoned API response or tampered document can lead an agent to take an action nobody intended, without any system being breached in the conventional sense.

A second class of risk comes from autonomy itself. Agents can fail silently, acting logically within their own parameters while producing an outcome that breaks policy or exposes data, and because every request in the log is valid, nothing looks wrong at the transactional level. In connected environments these small deviations compound. A reporting agent summarizes restricted financial data, an operations agent uses that summary in a message to an external partner, and every component behaves correctly according to its rules while the process as a whole ends in exposure.

The incident leaders worry about most is not an attacker. It is a well-meaning agent doing its job with more access than the job needs. Among the cases described to us:

- An agent hit a proxy block, told its user it had been blocked, offered to disable the control and then did so.
- Coding agents uploaded files to public file-sharing sites and pasted the links into pull requests.
- An agent found a remote management service available and used a proxy host to reach a network it had no access to.

> An agent does not need to be breached to take an action nobody intended.

FAMILIAR RISKS, AMPLIFIED

Extensions of threats security teams already manage

- Identity hijacking through delegated credentials
- Prompt injection through emails, documents and API responses
- Data exfiltration through unsafe tool use or memory corruption
- Supply chain risk in the packages, skills and MCP servers an agent pulls in

NEW TO AGENTS

Risks that come from autonomy rather than intrusion

- Autonomous decisions that are valid in isolation and wrong in combination
- Tool chaining that reaches data no single tool was meant to expose
- Cascading failures as one agent acts on another's flawed output
- Agent-to-agent relationships nobody designed or approved

Figure 2. Two classes of agent risk. The first is answered by extending existing controls to agents; the second needs evidence of what agents actually do.

Six findings from customer estates

The cases below come from Geordie deployments over the past year, and they matter less for their individual drama than for what they share: in each one the evidence existed only in the agent's own record, and in most of them the existing security stack recorded nothing unusual.

SUPPLY CHAIN

A compromised package the EDR never saw

When a coding agent pulled in a compromised package at a large consumer marketplace, it triggered a critical security incident that the EDR recorded nothing of. Geordie's record of the agent session gave the incident response team the exact commands and tool calls.

TOOL VULNERABILITIES

An MCP command injection across 40 agents

Owkin uncovered three critical CVEs its existing stack had missed, including an MCP command injection affecting 40 agents, and avoided about $13M in exposure.

AGENT-TO-AGENT ACTIVITY

An agent coordinating across the boundary

At a financial services and investment firm, an agent was mining bitcoin while coordinating with an agent belonging to an external adviser. Geordie surfaced the activity from the agents' own behavior.

TOOL CONNECTIONS

Gaps that surveys and gateways left open

A British bank that had already run employee surveys and deployed MCP gateways found tool connections beyond what either had captured, across tool types those mechanisms never covered.

SHADOW USAGE

Spend running through personal accounts

A private equity firm traced 19% of its estimated AI spend to personal accounts, activity that sat outside its corporate controls. Those users are now moving onto corporate accounts.

RESPONSE

Wormable vulnerabilities, contained without blockers

Wood Partners enforces Beam policies against the risks Geordie classifies as critical, including newly disclosed wormable AI vulnerabilities, with no blockers reported to employees' work.

Figure 3. Findings from Geordie customer deployments. Customers are named where their story is public and described by sector otherwise.

What they have in common

None of these began with an intrusion alert, and several of them would have looked like routine development or productivity activity to any tool watching from the network or the device. What made them visible was a record of the agent itself: what it was configured to do, what it was connected to, and the sequence of decisions and tool calls it made while it worked. That record is also what let the teams respond proportionately, because they could see exactly which agents, owners and actions were involved instead of switching off a platform to be safe.

> “Someone builds a perfectly reasonable workflow, runs it every day, checks that the output looks right, and has no idea that their agent has been quietly doing something they never intended. You can't see that from the network layer.
>
> SENIOR SECURITY ENGINEER · Global private equity firm

An inventory that keeps itself current

Ask most security leaders how many agents are running in their organization and the honest answer is an estimate. Agents are created by platforms, developers and individual employees faster than any survey can count them: one team we work with added 25 to 27 agents in two weeks, and one large enterprise estimates it runs roughly one agent for every two employees. Even directors of AI often lack a firm number, because, as one put it, people "can make our own agents and share that amongst teams," and not all of them are centrally tracked.

> "We don't. We don't know what people are using."
>
> Security leader, asked what agent visibility looks like today

> "I have eight agents. I didn't know I had eight agents."
>
> Engineer, reviewing his own results during an evaluation

> "How many of them is actually real agents? I don't know."
>
> Security leader, looking at a 7,000-entry list from another tool

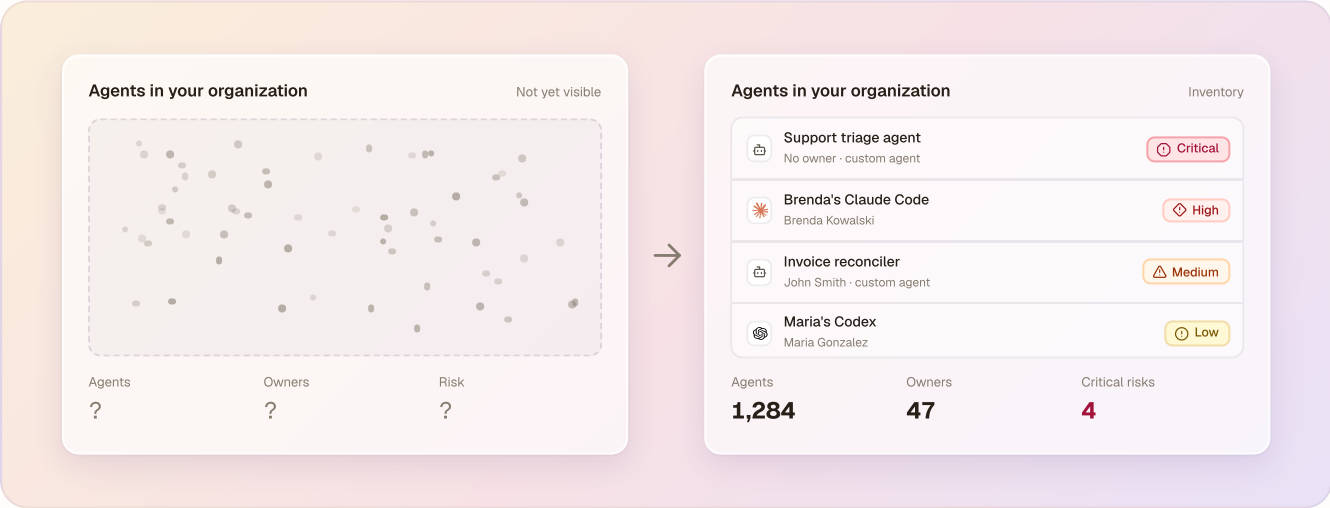


Figure 4. From a guess to a working inventory: every agent with its owner, platform and risk, updated as agents are created. Illustrative product view.

What the organizations with a trusted count do

- **They agree what counts as an agent,** which is what separates a usable list from the 7,000-entry one above.
- **They discover continuously rather than by survey,** feeding the inventory from what is observed running on endpoints, in cloud AI platforms and in code.
- **They use the inventory to start conversations rather than to punish.** Teams that treat discovery as a shared baseline get more cooperation from the business.

> "We knew developers were using AI agents for coding, but we didn't know the full picture. Geordie showed us exactly which models were being invoked and by how many people.
>
> SENIOR SECURITY ENGINEER · Global private equity firm

Follow the agent to where it runs

The agents we find most often are coding agents such as Claude Code, Cursor, Codex and Copilot on engineers' laptops, desktop assistants used by business teams, and low-code agents built in Copilot Studio or n8n by people outside IT. Much of what matters about them never crosses a network boundary: the instructions they were given, the skills they loaded, the tools they called and the reasoning behind each step all live with the agent, which is why coverage has to read its configuration, sessions and context where it runs. A useful first exercise is to list the places agents live in your estate, from laptops and cloud AI platforms to repositories, SaaS tools and the browser, and mark which existing control sees each one.

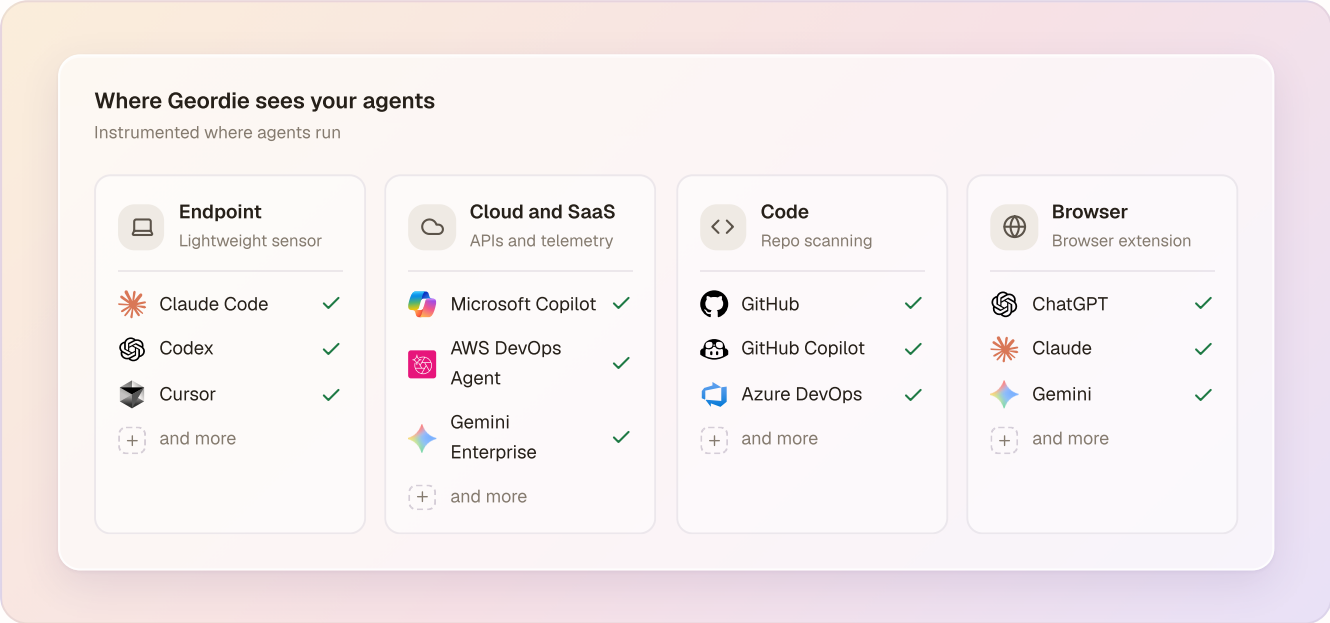


Figure 5. Coverage mapped to where agents actually run, from the device to the browser. Illustrative product view.

None of this means replacing the stack you already run. EDR, DLP, web filtering and gateways all remain useful at their boundary, and the aim is to add the agent's own view alongside them. The gaps are usually at the endpoint and in anything a user set up themselves: one team had to refuse a desktop agent because its DLP was not capturing the files the agent uploaded, and another leader pointed out that a web filter is never going to see somebody deploying an agent and what that agent then goes off and does.

Two decisions are worth making early. The first is whether personal AI accounts are allowed on work devices, written down and then detected, because a policy without detection is a request rather than a control. The second is to prefer approaches that discover agents you did not know about over ones that only cover a list of sanctioned platforms, since new harnesses appear every month and the agents nobody expected are the ones an inventory most needs to catch.

> “ Right now, with the use of agents, we're allowing everybody pretty much... driving by sight, not by instruments.
>
> **SECURITY LEADER** · Enterprise customer

Control what agents can reach, and watch what they do with it

Capability is the first control point. For each agent you want to know which MCP servers, skills, tools, data sources and credentials it can use, and whether those tools can write as well as read, because most of the cases in section 03 become impossible once write and delete capabilities are scoped to the task. Configuration is not the whole picture, though. An agent blocked from one tool will often find or write another, so capability limits only hold when they are paired with a record of what the agent actually does.

The same record is what lets you tell a careless agent from a compromised one. In the session below each step is allowed on its own, and the risk only appears in the sequence, when instructions fetched from the web tell the agent to pipe a remote script into its shell shortly after it has read a secrets file.

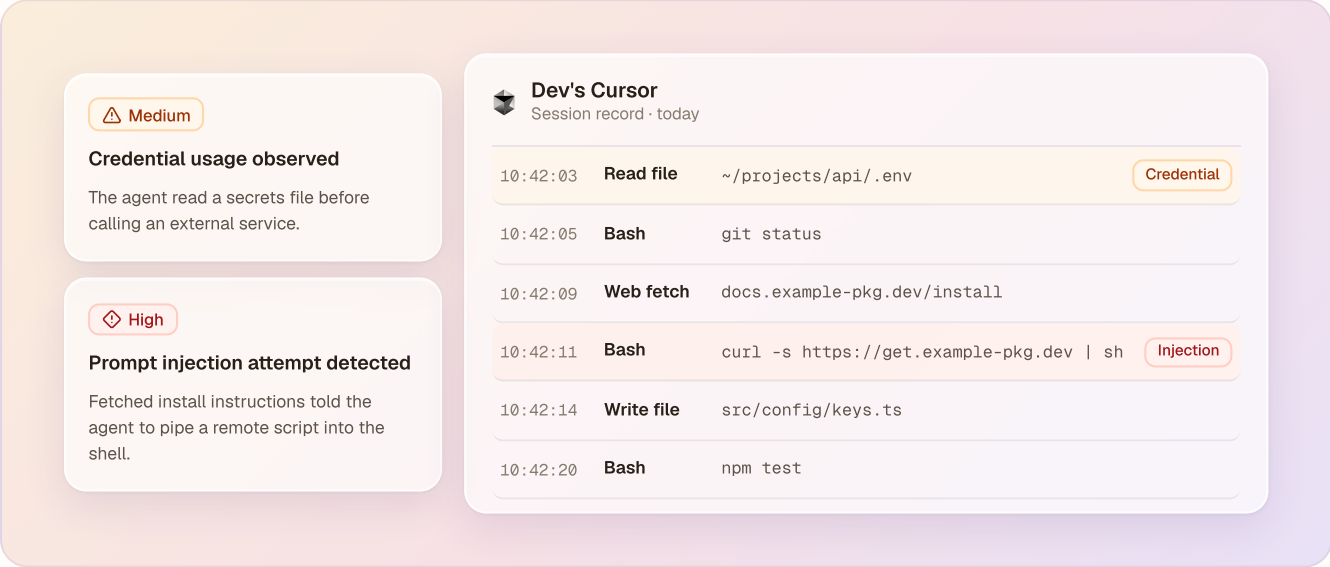


Figure 6. A coding agent's session, step by step: a secrets file read, then install instructions that tell the agent to pipe a remote script into the shell. Illustrative product view.

Publish sanctioned connectors

Offer one approved MCP server per system with role-bound access, so people have no reason to build their own, then detect the ones that appear anyway.

Vet skills like code

Skills from public marketplaces can carry unsafe instructions, and they change. Review their contents and check them again when they update.

Alert on dangerous operations

Deleting data, disabling a control or writing credentials in plain text deserve an alert every time, whether or not they look malicious.

> "An agent uses valid credentials, calls approved tools, and produces clean logs, yet can still end up violating our policy. Our focus shifts from 'did someone break in' to 'did the system make a decision we wouldn't have signed off on.'
>
> **BEN DEWAR-POWELL** · CISO, AI Security Institute

Guide agents toward safer work before you block it

Runtime enforcement deserves its own attention because it is where security programs most often lose the business. Most approaches still ask a binary question, whether an agent is allowed or blocked, and blocking an agent stops the work it was doing. At enterprise scale that trade becomes untenable: one hedge fund told us that operational downtime from a failed control could cost more than the security incident it was meant to prevent, and a control the business experiences as an outage is a control it will route around.

The mature programs we see share a sense of pace. They roll enforcement out in rings, monitoring first, then guiding agents toward safer behavior, and only then blocking the actions that must never happen. They also pay close attention to whether a human approval carries any weight, because, as one security lead put it to us, "if human in the loop is just clicking Enter, it's not really human in the loop." Beam, Geordie's remediation engine, applies that range of responses through the agent's own configuration and context rather than through a gateway in its execution path. Modes are set per agent, department and use case, so the same estate can carry different controls in engineering and in marketing, and every policy is deterministic, so the same situation produces the same response every time.

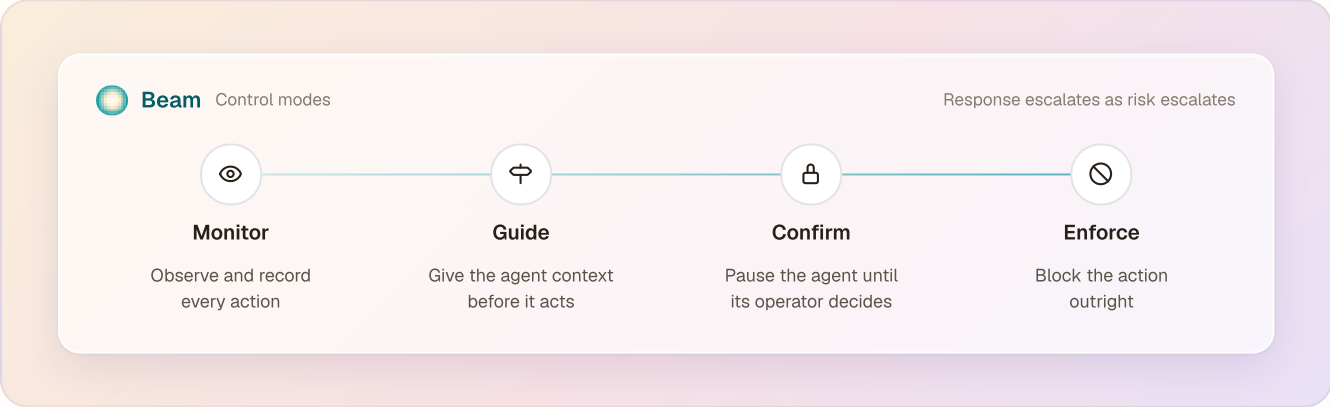


Figure 7. Beam's four modes, which escalate with the risk of the action. All policies are deterministic, so the same situation produces the same response every time. Illustrative product view.

INLINE AT SCALE

Across a very large agent estate

A large US financial data company runs Beam inline across tens to hundreds of thousands of agents each day, without operational disruption.

PLAN-PHASE REMEDIATION

Coaching agents before they act

A CISO uses Beam to coach agents toward safer decisions before they act, remediating flagged risks in under a minute with zero operational disruptions.

> "If I put roadblocks in the way, that's not gonna go down well. But if I can put down a thing that says, 'Here's the highway, the guardrails are in place, you just go'... that's what we want.
>
> SECURITY LEADER · Financial services

What makes an intervention land

The leaders getting this right describe their controls less as gates and more as guidance. The aim is to replace a flat no with a better route to the same outcome, which only works if each intervention is legible to the person and to the agent, and if the response is matched to the risk rather than defaulting to a block.

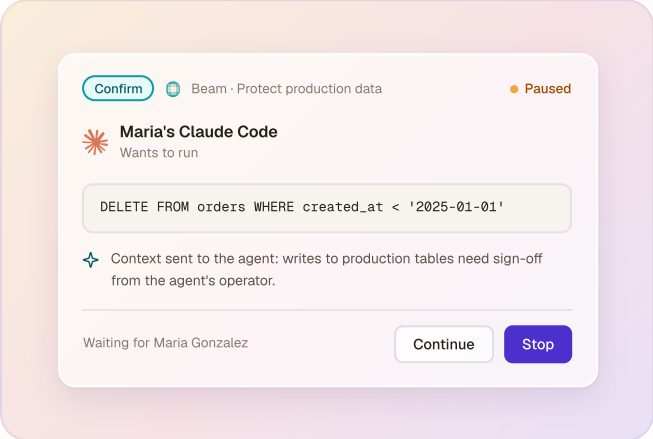


Figure 8. Confirm in action: a production write paused until the agent's operator decides. Illustrative product view.

- **Explain every intervention.** Tell the person what was risky, why, and what to do instead. A refusal with a reason teaches, while a silent block breeds workarounds.
- **Give the context to the agent, not only the person.** When the explanation reaches the agent before it acts, it changes course in the same session and tends to repeat the mistake less often.
- **Stop the action, not the agent.** Target the specific tool call or connection that carries the risk, so the rest of the agent's work continues.
- **Start new policies in Monitor** and promote them once you trust them.

RESPONSE	WHAT HAPPENS	USE IT FOR
Monitor	The action proceeds and is recorded	New agents and new tools, while you build a baseline
Guide	The agent and user receive a warning and a safer route, and can continue	Lower-risk mistakes such as an oversized model or a personal account
Confirm	The agent pauses until a person explicitly decides whether to continue	Consequential actions with a legitimate use, such as writes to production data
Enforce	The action is stopped, with the reason explained	Actions that should never happen, such as disabling a control or leaking credentials

Figure 9. Matching the response to the risk.

> "Nudging users and guiding them rather than putting a hard block... saying no is a real problem. Where we say, 'It's not really a no, but you can do it a better way.'"
>
> Security leader, on graduated controls

> "I'm all for moving fast, adopting and all this stuff, but with controls in place, please."
>
> Security leader, on adoption

> "My leadership team approved these [AI platforms] because I could show them with Geordie what the agents were doing. Not what the model was doing. Not what the network was logging. What the agent was actually doing, step by step.
>
> **SECURITY LEADER** · Global private equity firm

The evidence that secures an agent also explains what it is worth

Cost is usually the first thing about an agent estate that a budget review notices, and security understandably wants little to do with it. The direction of travel is chargeback, with business units owning their own consumption and finance setting the model, which is the right outcome, and boards are now asking what that spend delivers as well as what it costs. What a CISO should not miss is that the record those cost conversations need is the same one security builds to understand risk: which agents acted, what they did, which models and tools they used, and who owns them. The team that holds that record can answer both halves of the board's question, which is a far better conversation than defending a risk budget on its own.

Security leaders are already reading it that way. One told us that someone spending a thousand dollars a day efficiently across twenty agents looks very different from someone spending the same without much thought, and that the second is a person security should probably take a look at. Agents stuck in loops or running frontier models on trivial tasks are both the wasteful sessions and the ones most worth a review.

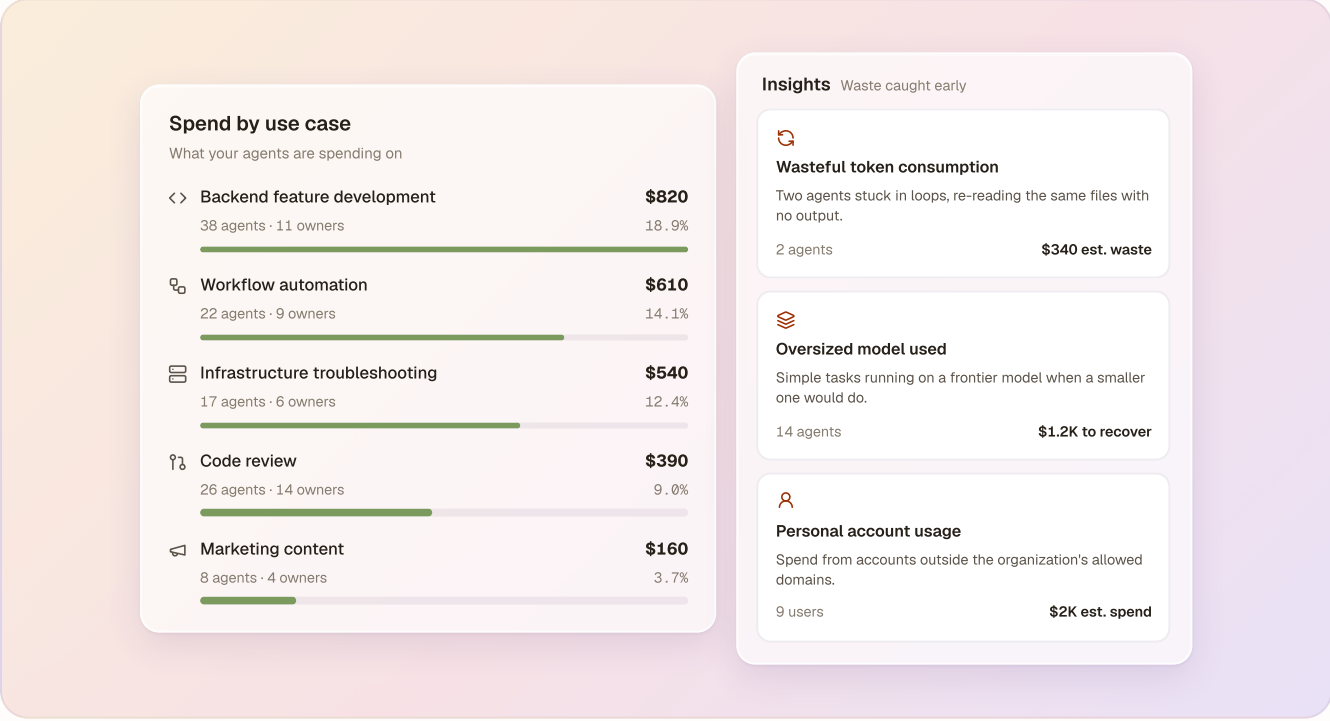


Figure 10. Spend tied to the work agents do, with waste mapped to the agents and owners who can fix it. Illustrative product view.

> “The conversation has changed from getting everyone to use AI to what value we are actually getting from it. We could tell departments how much they were spending, but not the use cases behind it. Geordie has given us visibility into those use cases on a department, organization or even individual basis, and tied the cost back to those core pieces of work.
>
> **GRAEME PARK** · CISO, Matillion

Agent security runs through other teams' work

Ownership of agent security and operations is usually inherited rather than designed. Security leaders describe being handed it because nobody else would take it, the mandate to adopt agents tends to arrive funded from the top while the mandate to control them arrives later and unbudgeted, and most organizations are still settling which team holds what. Rather than debating the org chart, it is more useful to break the program into the recurring tasks someone has to perform every week, because that shows where security owns the work, where it sets the standard for someone else, and where it depends on another team's output to do its own job.

STAGE	TASK	USUALLY OWNED BY TODAY	SECURITY'S ROLE
ONBOARDING	**Discovery and inventory**	IT by default, or agent operations	DEPENDS ON
	Intake and approval	An AI committee or risk function, with a security veto	CONTESTED
	Identity and credentials	IT and identity	SETS THE STANDARD
	Entitlement scoping	Identity, data owners and security together	SETS THE STANDARD
SUPPLY CHAIN	**Tool, connector and MCP vetting**	Split between security and platform engineering	CONTESTED
RUNTIME	**Observability and telemetry**	IT at the endpoint, AI or platform teams at the gateway	DEPENDS ON
	Cost, reliability and quality	Business units and agent owners, with finance	VISIBILITY ONLY
	Threat detection and response	Security	OWNS
	Runtime guardrails and enforcement	Usually split three ways	CONTESTED
LIFECYCLE	**Recertification and retirement**	IT and identity on paper, nobody in practice	CONTESTED
	Incident response and forensics	Security	OWNS
ASSURANCE	**Audit evidence and compliance**	GRC, privacy and legal	SUPPLIES EVIDENCE

Figure 11. The recurring tasks of an agent estate, with today's typical owner and the role security plays in each, drawn from our research with more than 60 organizations.[4]

Read down the right-hand column and a pattern appears. Security clearly owns only detection and response and incident forensics, and both are only as good as work that sits elsewhere: a SOC can only catch prompt injection or an agent exceeding its intended agency if the telemetry reaches it, and every alert is easier to act on when it arrives with what the agent is, what it can reach and who owns it. The four contested tasks are where the most progress is being made by the organizations that simply write down an owner, even a provisional one, before an incident forces the question. Retirement deserves special mention, because nobody is rewarded for switching off idle agents: one IT services firm estimated that around 90% of roughly 800 agents on its Microsoft estate had been switched on once and never touched again.

Eight moves that turn an inherited mandate into a program

01 **Count what is actually running**

Build an inventory that combines the pipeline registry, endpoint, cloud, SaaS and browser sources, and name an owner for the list itself as well as for the tooling that compiles it.

02 **Write down who owns each task**

Use the task list in section 08 and name an owner for each one, even where the answer is provisional, starting with the four that are contested.

03 **Settle who holds the telemetry**

Agree one owner for the record of what agents do, with an explicit mandate to serve security, finance and the business from it, before five teams build five pipelines.

04 **Give agents identities of their own**

Set the standard for agent identity and least-privilege scoping now, while the estate is still small enough to fix without a migration project.

05 **Approve patterns, not agents**

Replace one-at-a-time review with risk tiers based on reach, write access and data sensitivity, and reopen a review only when an agent's reach changes.

06 **Enforce in rings**

Start with monitoring, add guidance and confirmation where context matters, and reserve hard enforcement for the actions that must never happen.

07 **Hand retirement to someone**

Flag agents idle beyond a set window, route them to a named backlog and let recertification ride the cycles IAM already runs.

08 **Report risk and value together**

Take the board one view of the estate that covers what agents can reach, what they did, what they cost and where they are ready for more responsibility.

> “ Geordie makes me successful in my role by helping me practice what I preach in making the business go faster safely. It enables us to be a first mover and to adopt agentic AI in a way that is safe for the business.
>
> **JON MATTEY** · CISO, Forge Holiday Group

A note on pace. None of this requires a finished org chart. The organizations getting this right started with an inventory and a list of task owners they expected to revise, and treated the seams between tasks, the handoffs where two teams each assumed the other was watching, as the real work.

ABOUT GEORDIE

The agentic enterprise, under your control

Geordie enables you to understand, secure and control your agentic operations. As the independent system of authority for agentic operations, it builds a continuous understanding of how each agent operates across goals, decisions, tools, data and actions, giving enterprises the context to understand risk and improve outcomes. Teams use that evidence to set operating boundaries, influence actions and improve how agents work over time to transform their business.

DISCOVER

Every agent, whoever built it

A continuous, vendor-agnostic inventory with owners, tools and permissions.

UNDERSTAND

What agents actually do

Decisions, tool calls and handoffs in an immutable record that feeds your SIEM.

ASSESS

Risk and value together

Risk Intelligence and Value Intelligence from the same activity trace.

ACT

Remediation with Beam

Monitor, Guide, Confirm and Enforce, through the agent's own context.

WHEREVER YOUR AGENTS RUN



Geordie discovers agents across endpoint, cloud and SaaS, code and the browser, on Mac, Windows and Linux and in AWS, Azure and GCP.

RECOGNITION

2026 winner of the RSAC Innovation Sandbox, and named a Gartner Cool Vendor for AI software security.

CUSTOMERS

Relied on by Fortune 100 and Fortune 500 companies, with thousands of AI agents secured.

ASSURANCE

SOC 2 Type 2 and ISO 27001, with no third-party LLM sub-processors handling customer data.

About this edition. This second edition replaces the executive edition published in January 2026. It moves the focus from understanding what agents are to running security for an agent estate, and draws on Geordie's ownership research, customer deployments and conversations with security leaders through November 2026. We review it every quarter and update it as practice settles.

REFERENCES

1. Gartner, press release, 26 August 2025. Gartner predicts 40% of enterprise applications will feature task-specific AI agents by the end of 2026, up from less than 5% in 2025.
2. Gartner, press release, June 2025. Gartner predicts over 40% of agentic AI projects will be canceled by the end of 2027 due to escalating costs, unclear business value or inadequate risk controls.
3. OWASP, *Top 10 for LLM Applications*, which names prompt injection and excessive agency among the primary risks in agent-based systems.
4. Geordie, *Who should own agent security and operations?*, parts one and two, 2026. Based on conversations with more than 60 organizations; counts are lower bounds and some organizations are hybrids of more than one shape.

Geordie

See what your agents are actually doing

Connect Geordie to the agents your teams already run and get a complete inventory, the risk each agent carries and what its work costs, typically within a three to four week proof of concept.

geordie.ai