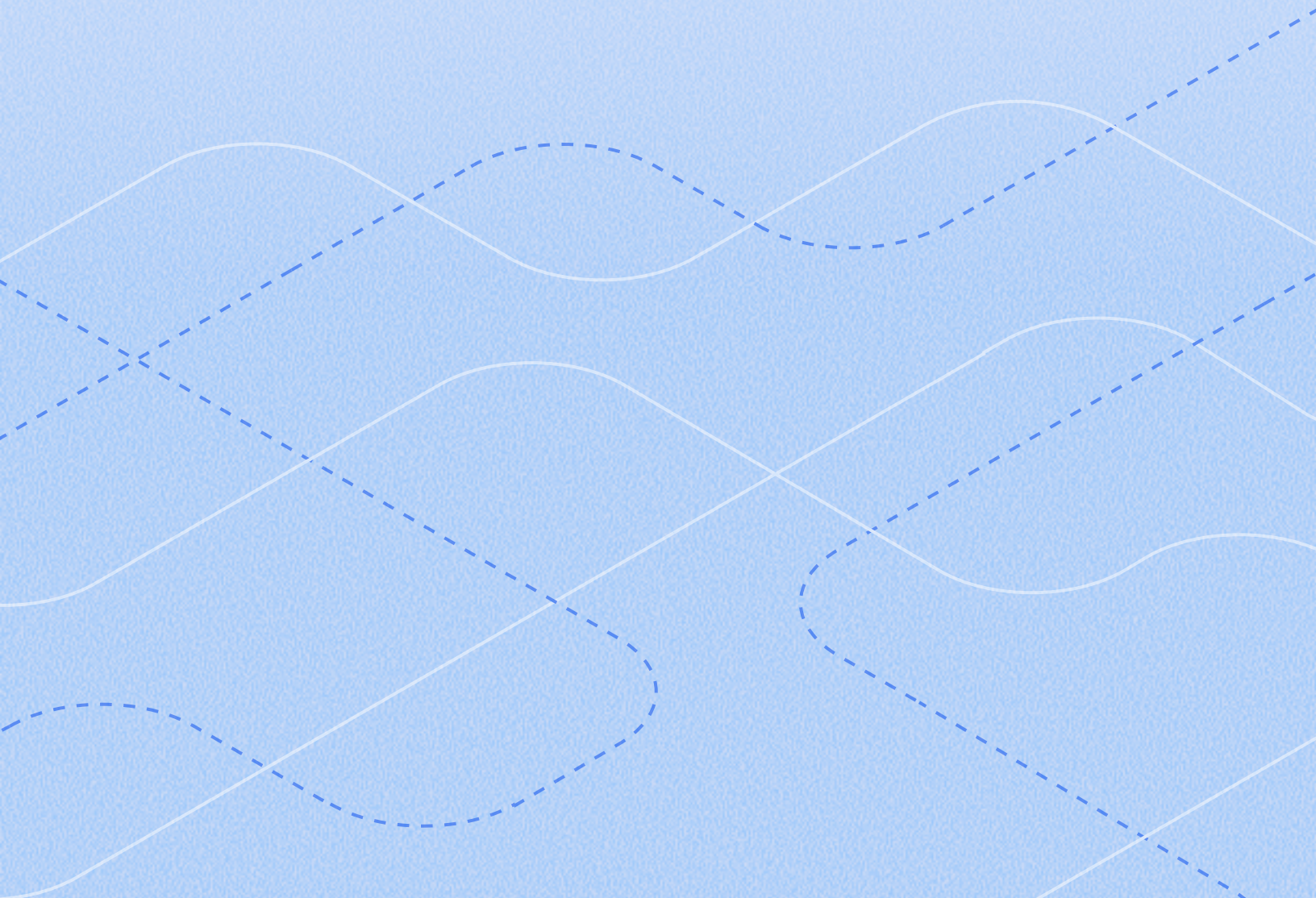
cyberhaven

Securing Data in the Agentic Enterprise

How Visibility, Observability, and Control at the Endpoint Make Agentic AI Safe to Adopt



Introduction

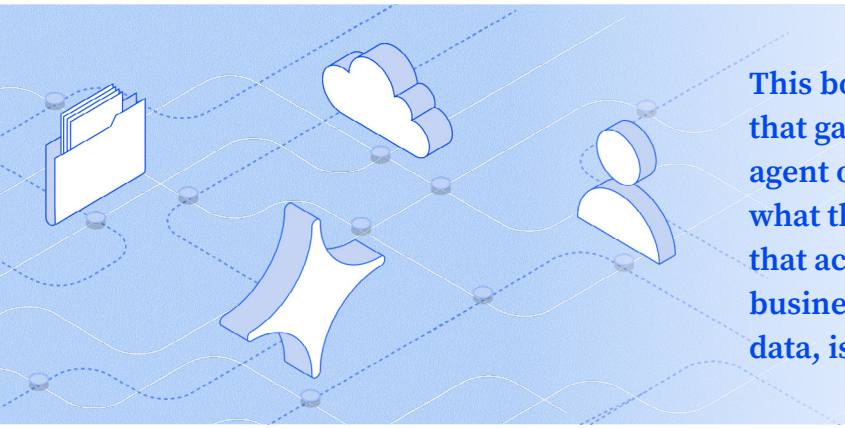
The Agentic Enterprise Is Already Here

Every enterprise is becoming an agentic one. Humans and autonomous AI agents now work side by side on the same data, inside the same workflows, often without a person reviewing each step. The question security teams face is no longer whether employees use AI. It is whether security teams can see which agents are running, understand what those agents are doing with sensitive data, and stop high-risk actions before they cause damage.

That shift changes the unit of security. Traditional programs were built to protect data points: a file, a record, a transaction. Agentic AI operates on workflows, sequences of actions that span systems, tools, and time. A program built to inspect single events misses everything that happens between them.

Point-in-time tools each see a fragment of this picture. Data security posture management (DSPM) maps where sensitive data lives. Legacy data loss prevention (DLP) inspects content as it moves. Cloud access security brokers (CASB) govern sanctioned applications. Endpoint detection and response (EDR) watches system behavior. None of them, on its own, sees the full workflow an agent executes.

Security teams can no longer simply say no. Blocking AI outright does not eliminate the risk. It eliminates the visibility, pushing agent use into the shadows where nothing is monitored at all. The gap between what security teams can secure and what the business is already doing is shadow AI, and it is growing faster than most programs can track.



This book lays out a practical framework for closing that gap. It rests on three pillars: visibility into every agent operating in the environment, observability into what those agents are doing with data, and controls that act on that understanding without stopping the business from moving. Protecting workflows, not just data, is the operating principle behind all three.

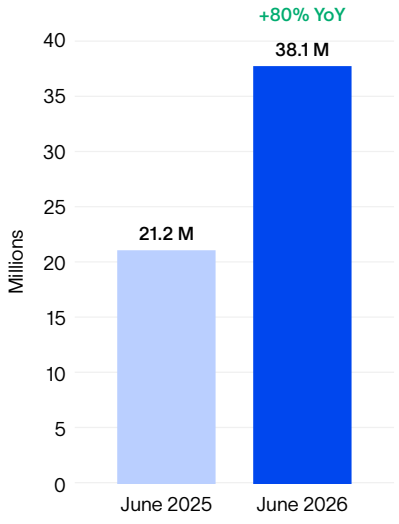
Chapter 1

The Adoption Curve Nobody Is Measuring Correctly

Enterprise AI adoption did not evolve gradually. It accelerated.

The first wave was familiar: employees using software-as-a-service (SaaS) generative AI (GenAI) tools such as ChatGPT, Copilot, and Gemini through a web browser. Security teams responded to this adoption with access policies, data classification rules, and browser-level controls. The governance question was tractable, if imperfect: who is using these tools, and what data are they sharing?

The second wave looks different. Developer adoption of AI coding assistants grew from 49.5% in December 2025 to a peak of 60.5% in May 2026, easing to 55.5% by June, as endpoint agentic AI adoption grew roughly six times faster than GenAI SaaS adoption over the same period. These are not applications accessed through a browser. They are agents installed directly on endpoints, operating inside filesystems, integrated development environments (IDEs), command-line interfaces (CLIs), and desktop automation frameworks, with direct access to code repositories, credentials, and production data.



Data movement events in and out of GenAI SaaS, June 2025 vs. June 2026.

The defining difference is not the interface. It is the threat model the interface creates. A browser-based GenAI interaction is episodic: a person sends a prompt, the model responds, the session ends. Agentic AI is continuous. An agent receives a task, plans a sequence of actions across multiple systems, invokes tools, reads and modifies files, calls application programming interfaces (APIs), and operates without a human reviewing each step.

The blast radius of a misconfigured or compromised agent is not a single response. It is a chain of autonomous actions that can span hours, systems, and data stores before any alert fires.

The governance questions that anchored first-wave security programs are not sufficient for this one. Knowing that a developer uses a coding assistant tells a security team nothing about what that assistant accessed, what it sent to an external model, or whether it operated within sanctioned parameters. Adoption rate is not risk. A small, fast-growing set of endpoint agents can carry the majority of an organization's exposure even with modest headcount adoption, because what matters is not how many people use an agent. It is what that agent can reach.

THE NUMBER TO REMEMBER

6x

Endpoint agentic AI adoption grew roughly six times faster than GenAI SaaS adoption between December 2025 and June 2026.

Source: Cyberhaven Labs, 2026 AI Adoption and Risk Report: Six-Month Update

Chapter 2

The Threat Landscape: Six Exposure Areas

Security practitioners doing threat modeling for agentic AI are working against a largely unmapped surface, one that keeps changing as users adapt and technology iterates. The risks are not hypothetical. The tooling to detect and respond to them is still catching up.

Enterprises deploying AI agents face six primary exposure areas.

01 Indiscriminate data access

Agents provisioned with broad access create a single point of failure. Without human judgment at the moment of retrieval, there is no contextual filter between an agent's task and the data it touches.

02 Shadow agent deployment

Developers and business units deploy agents outside formal security review, including locally installed tools, desktop automation frameworks, and open-source libraries with filesystem access and no governance footprint.

03 Loss of audit trails

Agents do not authenticate or log activity the way humans do. When an agent reads a file, transforms it, passes the output to another agent, and stores the result elsewhere, no native audit trail connects those events.

04 Sensitive data in AI pipelines

Personally identifiable information (PII), source code, financial records, and regulated data flow through agent workflows with no review of what is sent to an external model or stored in a vector database. Agents that aggregate across sources compound the risk.

05 Confidential data surfaced in outputs

Access controls on underlying data do not carry into the AI layer. An agent with read access to a filesystem can surface confidential information to users who would not otherwise have access to the source.

06 Compliance exposure

Agents optimize for task completion, not regulatory compliance. Without visibility into agent behavior, violations of data minimization, residency, or retention requirements go undetected until they become incidents.

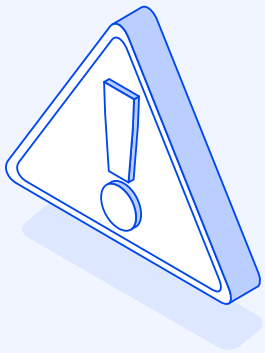
The Scale of Exposure

Nearly half of all organizational data is considered sensitive or confidential, yet only 32% of respondents to IDC's Data Security and Privacy Survey had more than 75% of their sensitive data mapped and monitored. AI agents operating across that unmapped landscape represent a risk multiplier, not a new risk category.

Data movement events into and out of GenAI SaaS applications grew **80% year over year** between June 2025 and June 2026, even as the number of sanctioned applications per enterprise declined slightly, evidence that fewer tools now carry far more sensitive traffic.

Source: IDC Spotlight, Rethinking Data Security and Insider Risk for Trusted AI Adoption, Jennifer Glenn, April 2026; Cyberhaven Labs

Shadow AI's Sharpest Edge: MCP



Shadow AI is the entry point for most other risks on this list. Roughly one in three employees access generative AI tools from personal accounts, and endpoint-based AI-native applications grew 509% year over year.

Model Context Protocol (MCP) servers are where that problem is compounding fastest: standing one up takes minutes, thousands of public implementations exist, and most enterprises have no inventory of what is connected to what. The sharpest risks are uncontrolled data access, indirect prompt injection through tool responses, and scope creep.

Four practices reduce the exposure: inventory every MCP server in use, apply least-privilege and read-only access by default, require human review for write or export actions, and treat every credential as a secret.

Source: Cyberhaven Labs

Chapter 3

Why Legacy Security Architectures Cannot Close the Gap

The failure of existing security tools to govern agentic AI is not a matter of configuration or coverage gaps. It is an architectural mismatch. Legacy security tools were built on four assumptions that agentic AI invalidates entirely: that data moves through controlled networks with predictable patterns, that humans are the primary actors triggering security-relevant events, that applications are known, static, and deployed through managed channels, and that risk events have detectable signatures that can be described in advance.

Agentic AI breaks every one of these assumptions. Data moves through agent pipelines, MCP servers, and APIs that bypass network-layer controls. The actor is not a human. It is software executing autonomously. Applications are installed locally by end users, not deployed through IT. The risk patterns that matter are behavioral, not signature-based: they emerge from sequences of actions across time, not individual events.

The Fundamental Category Error

API-based visibility is not endpoint visibility. A tool that monitors what data users send to a cloud AI service tells a security team about data at the boundary. It tells them nothing about what an agent did with a local file before it reached that boundary, what tools it called along the way, or whether a multi-agent workflow propagated a problem across systems before any alert fired.

EDR sees system behavior, such as process execution, file writes, and network connections, not data behavior. An alert that a process ingested a large number of files and transmitted data externally is a starting point for investigation, not an answer. Without knowing what data moved and where it went, the alert is nearly impossible to triage.

Legacy DLP was built for a world where data moved slowly through known channels. Static rules fire on volume and content patterns, generating alert fatigue that erodes security programs over time. They cannot distinguish a developer using synthetic test data from production PII being exfiltrated through the same channel.

Cloud-based and browser-extension tools are blind to the endpoint. Agents running locally in IDEs, CLIs, and desktop automation frameworks generate no telemetry, no alerts, and no audit trail as long as they never touch a monitored network endpoint.

Chapter 4

The Endpoint Is Where Risk Becomes Real

Most security programs have more visibility today than ever. Dashboards are full. Alerts are firing. Incidents are still happening. That is not a coincidence. Visibility and control are not the same capability, and the industry has spent years selling one while calling it the other.

Awareness without the ability to act does not reduce risk. It documents it. Once a security program can see a finding, the organization is accountable for it, whether or not the team can do anything about it. Visibility is about collection: pulling telemetry from endpoints, cloud environments, and applications and surfacing it somewhere a person can review it. Control is about action: sitting at the right layer of the stack to intervene, in real time, before data leaves the boundary where it belongs.

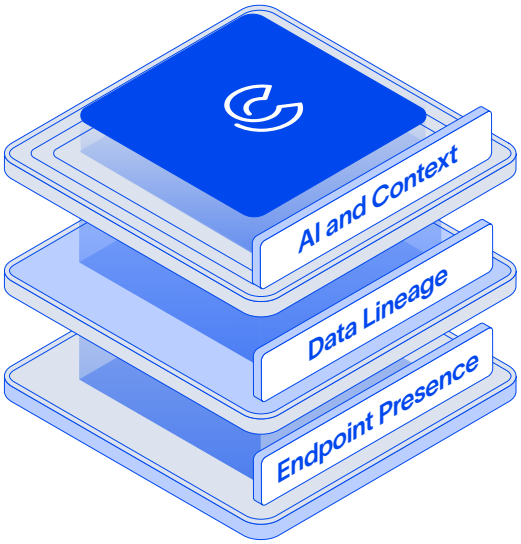
Data risk does not crystallize in storage. It crystallizes the moment someone, or something, acts on data: copies it, transforms it, moves it, pastes it somewhere it was never meant to go. That moment, and the context around it, is almost always at the endpoint, not in a cloud log or a network packet.

Visibility is a starting point. *Control* is the point.

Legacy DLP and standalone DSPM fail for different reasons and land in the same place. Legacy DLP relies on static rules built for a world where data moved slowly through known channels. It cannot distinguish a routine workflow from a genuine incident, so it generates the alert fatigue that erodes security programs over time. Standalone DSPM tells a security team where sensitive data lives across cloud environments, but it cannot say what a user did with that data the moment it left storage or which application it moved through. DSPM sees data at rest. The risk happens in motion.

Endpoint presence is the foundation the rest of the architecture depends on. It is what makes data lineage possible, because lineage cannot be built from snapshots. It has to be built from continuous observation of how data moves. And lineage is what gives artificial intelligence something real to reason over. Presence without lineage produces events a team cannot interpret. Lineage without AI produces a history a team cannot act on at scale.

Agentic AI raises those stakes further. Agents act on data without a person initiating each step, and that activity does not register in tools built for a human-driven model. The only way to see what an agent is doing with sensitive data is to have presence at the endpoint, where those actions actually execute.



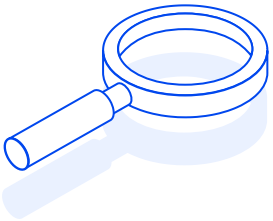
Removing any one layer breaks the architecture.

Chapter 5

The Three Pillars: Visibility, Observability, Controls

Governing agentic AI requires three capabilities that compound on each other. Visibility without observability means a security team knows what is running but not what it is doing. Observability without controls means the team can document risk but not prevent it. Controls without visibility and observability mean enforcement is blind, firing on patterns rather than context. The three pillars work together, and each depends on the others.

The endpoint is not just a device. It is where users and software act on data. Exfiltration, insider risk, and unsanctioned AI usage all crystallize at the endpoint, because the endpoint is where intent becomes action. Network-layer controls see the transmission. Endpoint controls see the decision.



Pillar One

Visibility, know what is running

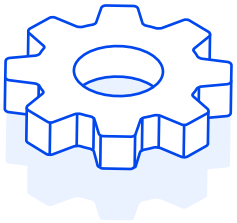
Effective visibility means continuously discovering and risk-scoring every AI agent, application, and MCP server in the environment, including agents running locally on endpoints that cloud-only tools cannot see.

- Continuous agent and MCP inventory across endpoints
- A real-time registry distinguishing sanctioned from shadow tools
- Risk scoring across multiple dimensions
- No manual cataloging or user-reported intake required

Pillar Two

Observability, understand what agents are doing

Prompt-level inspection is not sufficient, because violations do not always appear in a single message. An agent may access sensitive files in one step, pass that data to an external tool a few steps later, and trigger a compliance violation further downstream, with no individual action appearing anomalous on its own. Observability means reconstructing the full execution lifecycle: which data was accessed, which tools were invoked, what actions resulted, and how each step connects to the next.

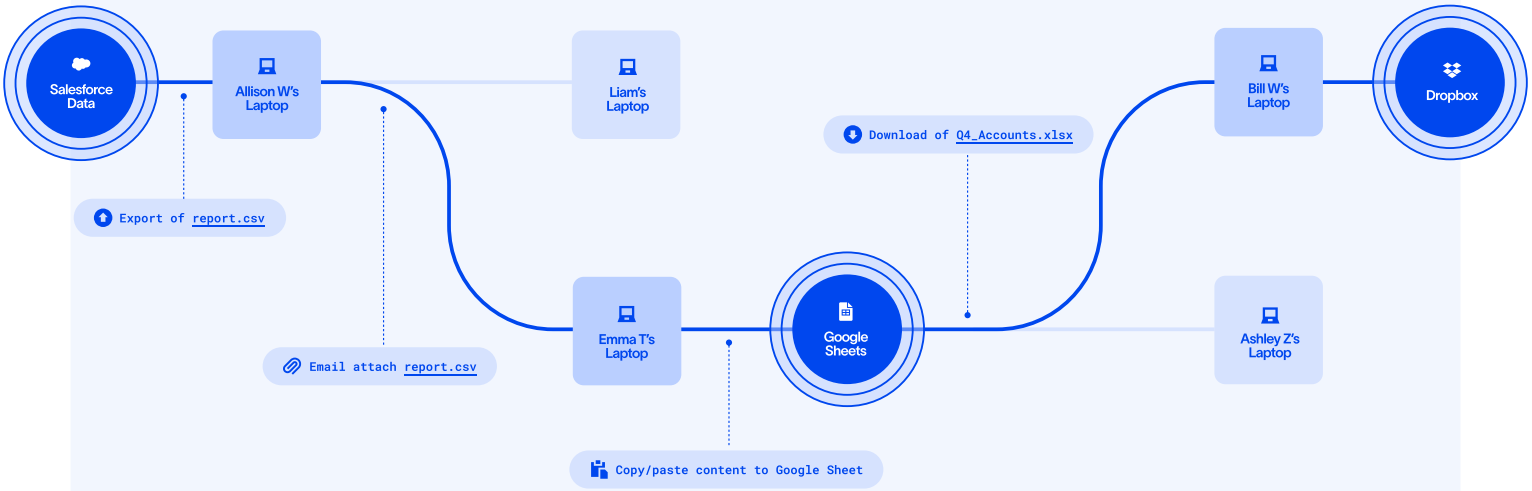


Pillar Three

Controls, enforce guardrails at the moment of execution

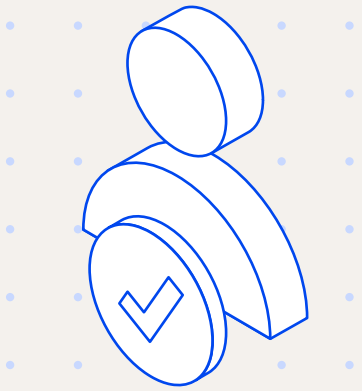
Block-first controls generate alert fatigue, drive shadow agent adoption further underground, and create operational drag without reducing risk. An agent blocked from completing a task does not disappear. The user finds a workaround. Effective controls are context-aware. They distinguish a developer using synthetic test data from an agent exfiltrating production PII through the same channel, and they coach, warn, or redact rather than block outright.

Data lineage is the capability that connects individual events into a coherent picture of risk. It tracks how data is created, copied, modified, and shared over time, preserving the chain of custody as data moves between files, applications, agents, and external destinations. Without lineage, security teams see events in isolation: a file was accessed, data was transmitted, an alert fired. With lineage, they see context: this data originated in a financial system, was read by an agent connected to an external model, was transformed and stored in an embedding database, and later surfaced in a user's output.



Principle of Least Privilege Access

Overly broad permissions are the primary reason agentic AI incidents are so damaging when they occur. An agent provisioned with the minimum access required to complete its task cannot exfiltrate what it cannot reach. Enforcing least privilege at the agent level requires the same infrastructure as enforcing it for human users: continuous inventory, behavioral monitoring, and policy enforcement at the point of access.



Chapter 6

Governance Needs a New Enforcement Model

> "The security question is now: What did the agent read, aggregate, and send, and did a human authorize that specific action?"

Governance for agentic AI has to ask a different question than the one first-wave programs were built to answer. A workable enforcement model has three parts, in a specific sequence: discovery, observability, and control.

Sequence matters because each stage produces the input the next stage needs. Discovery without observability is a list a security team cannot interpret. Observability without control is a record they cannot act on. Skipping a stage does not save time. It produces a gap the next stage cannot close on its own.

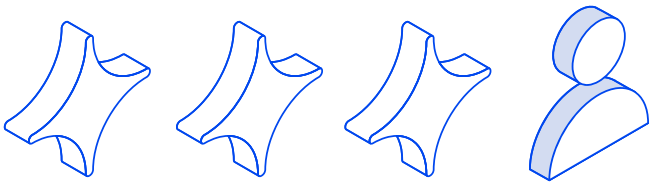
Governance also starts above the tooling layer. Before any platform decision, the organization has to decide its risk appetite and prioritize which assets matter most. A security team that skips this step ends up applying uniform policy to workflows that carry very different levels of risk, over-restricting the ones that matter least and under-protecting the ones that matter most.

What Automation Can't Take Off Your Plate

Regulators and boards look for the person, not the algorithm. When SolarWinds and Uber faced scrutiny after security incidents, the accountability question ultimately landed on named individuals in the chief information security officer (CISO) role, not on the tools those individuals had deployed. That precedent holds for agentic AI. Automating a decision does not automate away who answers for it.

The clearest way to think about this is the parallel to general counsel. Legal teams use software throughout their work, but a human attorney still signs the filing. Security teams are moving toward the same model with AI. The agent can execute the task. A person still has to authorize the action, and the organization still needs a record of who did.

That has a direct operational consequence: oversight infrastructure and a defensible data trail have to exist before an incident, not after. A security team that can reconstruct what an agent did, when, and under whose authorization is in a fundamentally different position during an investigation than one piecing the story together from fragmented logs after the fact.



The agent executes. *A person* still signs.

Chapter 7

The Checklist: Categories to Audit Now

A security team assessing its readiness for agentic AI can work through six categories, mapped to three stages of the data lifecycle: input, processing, and output.

Input

- ☐ **Discovery:** Can the team enumerate which AI agents are running on endpoints, including locally installed tools that do not appear in the sanctioned SaaS inventory?
- ☐ **Access:** Are agent permissions scoped to the minimum required for the task, with no standing access beyond it?

Processing

- ☐ **Data lineage:** Can the team trace what happens to a file after an agent reads it, including whether it is stored in embeddings or sent to an external model?
- ☐ **Third-party risk:** Does the team have visibility into which MCP servers and external tools agents connect to, and who approved those connections?

Output

- ☐ **Monitoring:** Can the team differentiate between synthetic test data and production PII moving through agent workflows?
- ☐ **Incident response:** Can the team reconstruct a complete event chain for an agent-related incident without relying on manual log correlation?

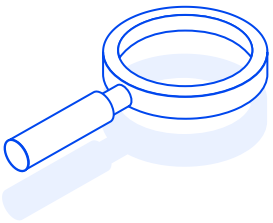
Each negative answer corresponds to a gap in one of the three pillars. The gaps compound: an organization without agent visibility cannot build observability, and one without observability cannot enforce context-aware controls. The program is only as strong as its weakest foundation.

Conclusion

Security as the Accelerant

The opportunity in front of security teams is not to be the function that says no. It is to be the one the business credits when it moves faster because the risk has already been handled.

Cyberhaven built its approach around this framework from the start.

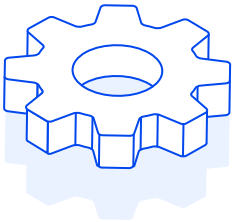


Visibility

Cyberhaven continuously discovers every AI agent, application, and MCP server across the enterprise, including endpoints and SaaS environments, and assigns a risk score to each with no customer configuration required.

Observability

Cyberhaven reconstructs the full execution lifecycle of AI agent interactions, tool calls, data access, and multi-turn conversation context, surfacing violations that only emerge across multiple turns.



Controls

Runtime policy guardrails stop high-risk data movement, redirect users to sanctioned tools, and coach employees in real time with plain-English explanations, enforcing governance without eliminating productivity.

Key Differentiators

- → **Endpoint plus cloud coverage.** Browser-extension and network-layer tools are blind to AI agents running locally on endpoints. Cyberhaven sees them.
- → **Conversation-level understanding.** Most tools inspect one message at a time. Cyberhaven watches the entire conversation, every step an agent takes, every file it touches, every tool it calls, including multi-agent workflows where a problem in one agent can propagate silently to the next.
- → **Data lineage as investigation context.** Every other agentic AI security tool reports what an agent did. Cyberhaven reports what an agent did to which data, where that data came from, and where it went next. That is the difference between an alert and an investigation.