cyberhaven

Governing the Autonomous Enterprise

A *Security* Framework for Agentic AI

How visibility, observability, and controls at the endpoint unlock safe AI adoption

AI Has Changed. Security Hasn't.

AI has crossed a threshold that security teams can no longer treat as a future concern. The question is no longer whether employees are using AI tools, but it is now whether security teams have the visibility to know which agents are running, the observability to understand what those agents are doing with sensitive data, and the controls to stop high-risk actions before they create damage or data leaks.

The shift from GenAI SaaS tools to agentic AI represents a qualitative change in the threat model. AI agents maintain state, operate continuously, and execute multi-step workflows across enterprise systems without human checkpoints. They access data programmatically, invoke external tools via model context protocol (MCP) servers, and act at a speed that overwhelms security programs calibrated to human-paced behavior.

Legacy security architectures were not designed for this new era. Endpoint detection and response (EDR) tools see system behavior, not data behavior. Legacy data loss prevention (DLP) fires on volume, not context. Early cloud-based AI security tools are blind to agents running locally on endpoints. The result is a rapidly expanding attack surface that lacks adequate coverage.

The time has come for a new security model, one organized around three core pillars: Visibility into every agent and connector in the environment, observability of agent behavior across full execution threads, and runtime controls that enforce policy without blocking productivity.

Built on a foundation of data lineage, this framework gives security teams the forensic context needed to govern agentic AI safely and accelerate adoption with confidence.

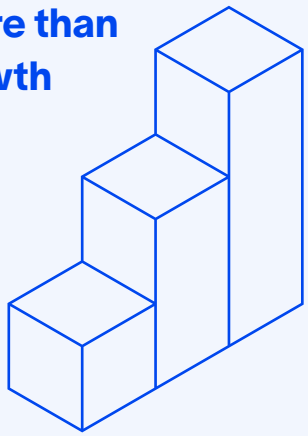
Section 01

The Second Wave of AI Adoption and Why It Is Different

Enterprise AI adoption did not evolve gradually, it accelerated. The first wave was familiar: employees using SaaS-based GenAI tools like ChatGPT, Copilot, and Gemini through web browsers. Security teams responded with access policies, data classification rules, and browser-level controls. The governance questions were tractable, if imperfect: *Who is using these tools, and what data is being shared?*

The second wave is different. Endpoint-based AI agents grew 276% over the past year, more than triple the growth rate of GenAI SaaS tools. By December 2025, roughly half of all developers (49.5%) were using desktop-based coding assistants, up from approximately 20% at the start of the year. These are not applications accessed through a browser, but instead they are agents installed directly on endpoints, operating within filesystems, IDEs, CLIs, and desktop automation frameworks, with direct access to code repositories, credentials, and production data.

Endpoint-based AI agents grew 276% over the past year, more than triple the growth rate of GenAI SaaS tools.



The defining difference is not the interface. It is the threat model the interface creates.

Traditional GenAI interactions tend to be episodic in nature. For example, a user sends a prompt, the model responds, the session ends. Agentic AI, however, is continuous. An agent receives a task, plans and executes a sequence of actions across multiple systems, invokes tools, reads and modifies files, calls APIs, and operates without human review at each step.

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

The governance questions that anchored first-wave security programs are no longer sufficient to secure this ongoing wave. Knowing that a developer is using a coding assistant tells you nothing about what that assistant accessed, what it sent to an external model, or whether it is operating within sanctioned parameters. These new questions require new AI security infrastructure.

Section 02

The Agentic AI Threat Landscape

Security practitioners doing threat modeling for agentic AI are working with a largely unmapped surface that is changing as users adapt and technology iterates. The risks are not hypothetical, but the tooling to detect and respond to them is nascent.

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, such as locally installed tools, desktop automation frameworks, open-source libraries, with filesystem access and no governance footprint.

03 **Loss of audit trails:** Agents don't authenticate or log activity the way humans do. When an agent reads a file, transforms it, passes output to another agent, and stores the result elsewhere, there is no native audit trail connecting those events.

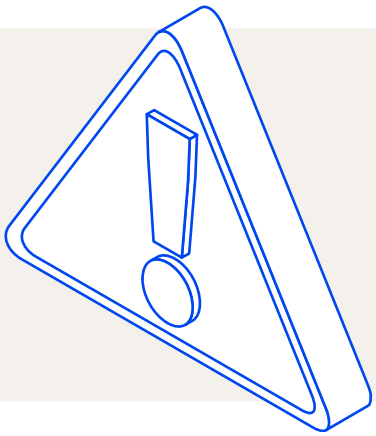
04 **Sensitive data in AI pipelines**: PII, source code, financial records, and regulated data flow through agent workflows with no review of what is sent to an external model or stored in a vector database. Agents that aggregate across sources compound the risk.

05 **Confidential data surfaced in outputs:** Access controls on underlying data don't 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.



Section 03

Why Legacy Security Architectures Cannot Close the AI Security 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.

01 Data moves through controlled networks with predictable patterns.

02 Humans are the primary actors triggering security-relevant events.

03 Applications are known, static, and deployed through managed channels.

04 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 API calls 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. And the risk patterns that matter in agentic AI 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 you about data at the boundary. It tells you 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.

Why Point Solutions Fall Short

EDR sees system behavior (e.g. process execution, file writes, 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.

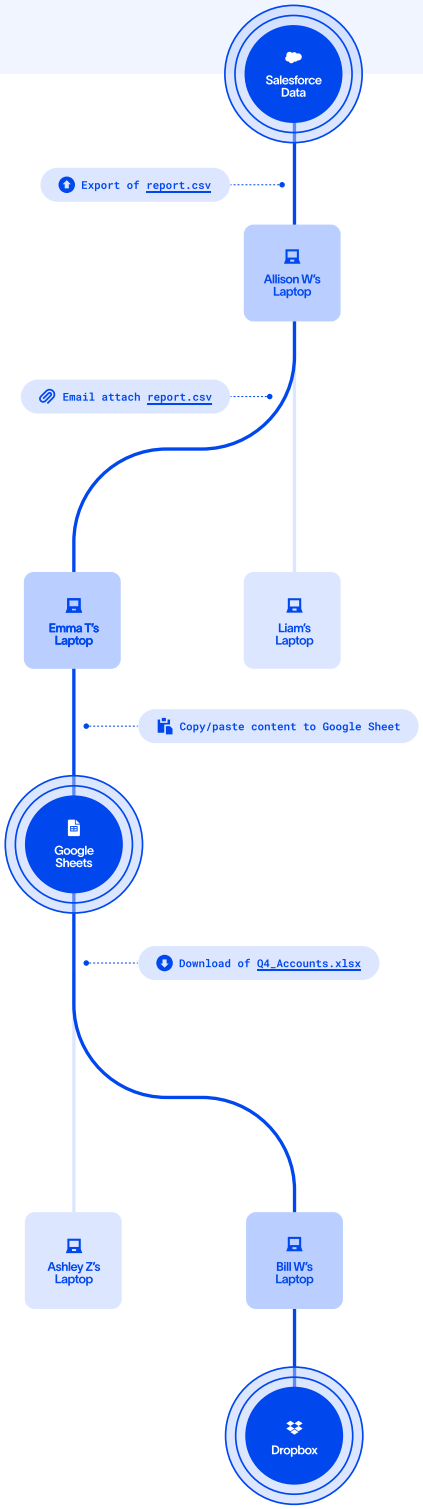
Legacy DLP was built for a world where data moved slowly through known channels. Static rules fire on volume and content patterns, generating alert fatigue that erodes security programs over time. They cannot distinguish a developer using synthetic test data from production PII being exfiltrated through 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 don't touch a monitored network endpoint. Developer laptops are the most common deployment environment for agentic AI, and the largest blind spot in current architectures.

The problem is not that individual tools are weak. It is that they were designed to operate independently on a threat model that no longer reflects how enterprise data moves.

Section 04

The Endpoint Is the New Control Plane for AI



The endpoint is not just a device. It is the system 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. A user who decides to copy a file to a personal cloud drive does that at the endpoint. An agent that reads a credentials file and transmits its contents to an external model does that at the endpoint. Network-layer controls see the transmission. Endpoint controls see the decision.

This distinction matters more as AI agents become the primary actors in enterprise data workflows. Agents increase both the volume of data being handled and the speed at which it moves. They operate continuously, often across multiple concurrent tasks, and they do not pause for human review. Security programs calibrated to human-paced behavior, with analysts reviewing alerts on a 24-hour cycle, are structurally mismatched to the threat model that agentic AI creates.

Data Lineage Is the Connective Tissue

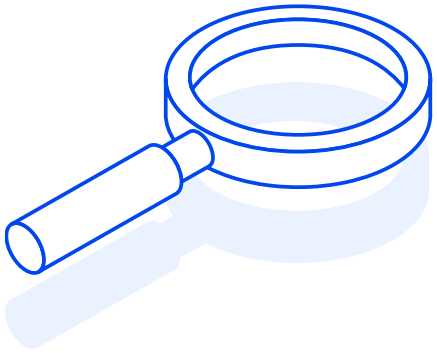
Data lineage is the capability that connects individual events into a coherent picture of risk. It tracks how data is created, copied, modified, and shared over time, preserving the chain of custody as data moves between files, applications, agents, and external destinations. Without lineage, security 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 with an MCP connection to an external model, was transformed and stored in an embedding database, and subsequently surfaced in a user's output.

The difference between an alert and an investigation is context. Data lineage provides that context at scale, automatically, without requiring analysts to manually reconstruct event chains from fragmented system logs. This is what enables precise enforcement decisions, whether the question is whether to block an agent action in real time or whether to escalate an incident after the fact.

Section 05

A Framework for Agentic AI Security

Governing agentic AI requires three capabilities that compound on each other. Visibility without observability means you know what is running but not what it is doing. Observability without controls means you 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.



Pillar One: Visibility

Know what is running

The same dynamic that produced shadow IT is now producing shadow agents. Without a systematic inventory, security teams are governing a subset of the environment, hoping the unmonitored portion does not cause an incident.

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

What good looks like:

- Continuous agent and MCP inventory across endpoints
- Real-time registry distinguishing sanctioned vs. shadow tools
- Risk IQ scoring across 5 dimensions
- No manual cataloging or user-reported intake required

Pillar Two: Observability

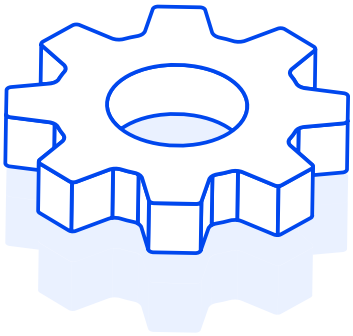
Understand what agents are doing

Prompt-level inspection is not sufficient. Violations do not always appear in a single message. An agent may access sensitive files in step three of the process, pass that data to an external tool in step five, and trigger a compliance violation in step seven, all with no individual action appearing anomalous in isolation.

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.

What good looks like:

- Data lineage tracked per agent action
- Full conversation thread and workflow analysis
- Multi-agent correlation across connected workflows
- Telemetry architecture built for agents, not human actors



Pillar Three: Controls

Enforce guardrails at the moment of execution

Block-first controls generate alert fatigue, drive shadow agent adoption underground, and create operational drag without reducing risk. An agent blocked from completing a task does not disappear from the environment, instead the user often 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. They block, warn, or redact based on what is actually happening.

What good looks like:

- Contextual coaching over blunt blocking
- Warn with option to revise when prompts contain sensitive data
- Redact specific elements rather than blocking entire interactions
- Plain-English policy explanations to improve compliance over time

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.

Section 06

Why Endpoint DLP Is the Foundation of AI Security, Not an Add-On

Traditional DLP was designed for a world where data moved more slowly through more predictable paths. It operates on content inspection: rules that fire when a file containing a credit card number is emailed to an external address, or when a large volume of documents is uploaded to a personal cloud drive. That model is inadequate for agentic AI, where data moves through agent pipelines at machine speed, across systems that static rules cannot enumerate in advance.

AI-native endpoint DLP is different in kind. It operates at the point of action, with the behavioral context required to distinguish routine work from genuine risk. It does not fire on content patterns alone. It fires on the combination of what data is involved, who or what is accessing it, what the action is, and what the intent context suggests. That combination is only available at the endpoint, where all four signals converge.

Three engineering requirements that separate programs that work from those that quietly fail:

1. Uniform behavior across environments

A program that works on Windows but not macOS, or on managed devices but not BYOD, creates gaps that shadow agents exploit. Consistent enforcement requires an architecture that operates across the full diversity of real enterprise environments.

2. Stability at scale

A security tool that degrades system performance gets disabled by users and IT teams. Endpoint DLP for AI environments must run continuously with minimal overhead because the agents it monitors never stop running.

3. True data lineage

Seeing that data moved is not the same as understanding what it means. Lineage connects data actions, artifact histories, and system events into a context graph that makes enforcement precise. Without it, controls can fire on the fact that data left the endpoint, but cannot reason about why or whether the risk was real.

Organizations that deploy these capabilities from a unified platform will be better positioned for AI at scale. The prerequisite is an endpoint foundation capable of handling the volume, speed, and complexity of agentic AI data movement.

Section 07

Governing Agentic AI Safely

Enterprises deploying AI agents have three options for managing the associated risk. The first is to block AI usage broadly: deny access to unapproved tools, restrict agent frameworks, and enforce a restrictive posture. This approach does not work. It drives adoption underground, accelerating shadow agent deployment while eliminating visibility. The agents do not go away. The governance does.

The second option is to manage AI risk through a proliferating set of rules: content filters, destination blocklists, application allowlists. This is the whack-a-mole approach, and it fails for the same reason that traditional DLP fails. Rules cannot keep pace with the rate at which new agent frameworks, MCP servers, and AI tools are deployed. Each new tool requires a new rule. Alert fatigue accumulates. Exceptions accumulate. The program erodes.

The third option is to focus on the data. Rather than attempting to enumerate every AI tool and write a rule for every risk pattern, organizations that build security around data lineage can make enforcement decisions based on context: what data is involved, how sensitive it is, what the agent is doing with it, and whether that action is consistent with policy. This approach scales with AI adoption rather than against it.

A Practical Readiness Checklist

Organizations assessing their current posture for agentic AI security should work through the following questions:

- ☐ **Can you enumerate which AI agents are running on endpoints, including locally installed tools that do not appear in your SaaS inventory?**
- ☐ **Can you trace data lineage when an agent reads a file and stores content in embeddings or an external model?**
- ☐ **Can you differentiate between synthetic test data and production PII in agent workflows?**
- ☐ **Can you reconstruct a complete event chain for an agent-related incident without relying on manual log correlation?**
- ☐ **Can you enforce context-aware policy at the agent level without blocking legitimate use cases?**

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

Conclusion

Cyberhaven's approach

The Cyberhaven AI & Data Security Platform delivers agentic AI security through the AI Security module, built on the same Data Lineage layer that powers Cyberhaven's DLP, DSPM, and IRM capabilities. Agentic AI security was not bolted onto a legacy platform. The endpoint has always been the foundation.

How the Platform Delivers on Each Pillar

Visibility

Cyberhaven continuously discovers every AI agent, application, and MCP server across the enterprise, including endpoints, SaaS environments, and MCP servers, assigning Risk IQ scores across five dimensions with no customer configuration required. Applications are categorized as Sanctioned, Unsanctioned, Tolerated, or Restricted. This is the only approach that surfaces shadow agents running locally on endpoints that cloud-only tools cannot reach.

Observability

Cyberhaven reconstructs the full execution lifecycle of AI agent interactions: tool calls, data access, reasoning paths, and multi-turn conversation context. By analyzing full conversation threads rather than isolated prompts, the platform surfaces violations that only emerge across multiple turns and enables forensic-grade context for every AI interaction.

Controls

Runtime policy guardrails stop high-risk data movement, redirect users to sanctioned tools, and coach employees in real time with plain-English risk explanations and links to applicable policy. Users who submit prompts containing sensitive data receive a warning with the option to revise or proceed, enforcing governance without eliminating productivity.

Key Differentiators

Endpoint plus cloud coverage. Browser-extension and network-layer vendors are blind to AI agents running locally on endpoints. Developer laptops, IDEs, CLIs, and desktop agent frameworks are invisible to them. 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 tells you what an agent did. Cyberhaven tells you what an agent did to which data, where that data came from, and where it went next. This is the difference between an alert and an investigation.

Ready to see it in action?

Cyberhaven's Agentic AI Security gives security teams the visibility, observability, and controls needed to govern AI agents operating across the enterprise, from endpoints and browsers to MCP servers and SaaS integrations. Visit cyberhaven.com to request a demo.

Request a demo

About Cyberhaven

Cyberhaven protects sensitive data wherever it lives and goes. Built for the AI era, Cyberhaven's unified data security platform combines DSPM, Data Loss Prevention, Insider Risk Management, and AI Security with deep data lineage. Cyberhaven helps organizations stop data loss, reduce insider risk, and enable AI adoption securely, without slowing their business. Learn more at cyberhaven.com.

This whitepaper references data from the IDC Spotlight: **Rethinking Data Security and Insider Risk for Trusted AI Adoption**, written by Jennifer Glenn, Research Director, Information and Data Security, April 2026.