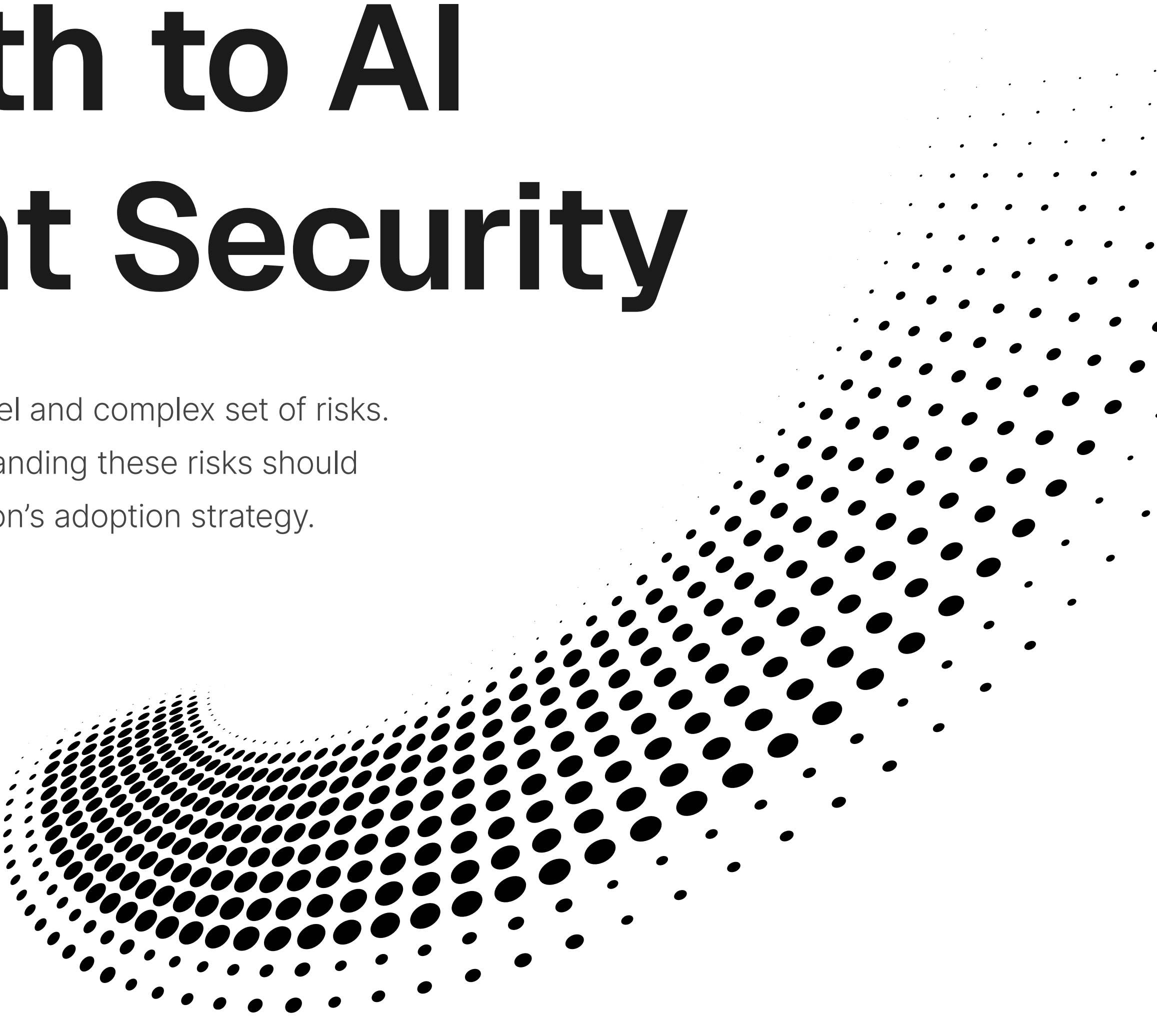
A Path to AI Agent Security

AI agents present a novel and complex set of risks. Identifying and understanding these risks should underpin the organization’s adoption strategy.



INTRODUCTION:

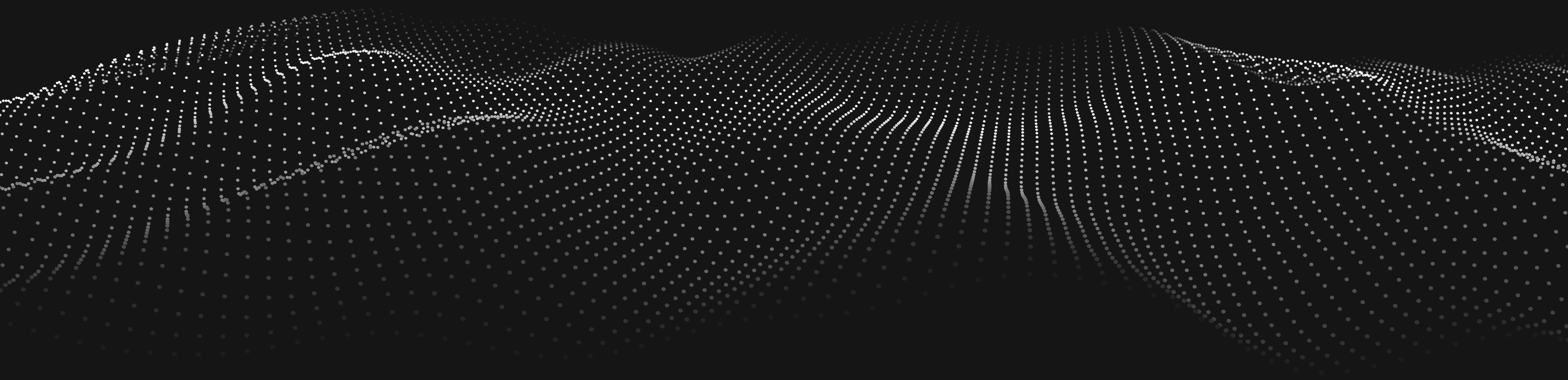
Standing firm while the world shifts under your feet

Securing AI agents represents a monumental shift for security and governance teams - even for those organizations that have already embraced a strategy for AI adoption. In the initial wave of AI adoption, security and governance concerns centered on data leakage or misuse from user interaction with LLMs, prompt protection against adversarial attacks, insufficient privacy protection and appropriate usage - whether for third party AI applications or custom built applications.

By contrast, AI agents can perform actions with varying degrees of autonomy and agency, use inference to determine what actions to perform or code to execute, and access datasets within different contexts - production environments, no-code tooling, and on shared infrastructure. This emerging category of software flows establish a new attack surface that existing methodologies or even repurposed low code security tools can't fully address.

This white paper discusses a framework, guiding principles and baseline technical controls to help organizations manage the balance and trade off between the productivity and the business gains from implementing AI agents, and the relative risks.

We outline what constitutes an agent, offer an initial taxonomy, and introduce key pillars for a practical strategy for the agentic landscape - as well as controls for detection of the systemic risk of context injection attacks for agentic runtimes.



LEVEL SET:

What is an AI agent?

The level of interest and engagement with AI technologies within organizations has followed an unprecedented trajectory. From initial adoption of LLMs in early 2024, enterprises have moved decisively to embrace AI, as well as stand up AI steering committees to define and execute strategies for leveraging AI. Agents and agentic AI are broadly seen as the next wave of enterprise adoption.

It’s important to differentiate between agents in the LLM era from “agent washing” - or rebranding of no-code and low-code platforms as agentic.

So what is an agent?

Gartner defines AI agents as “autonomous or semiautonomous software entities that use AI techniques to perceive, make decisions, take actions and achieve goals in their digital or physical environments.”

By comparison, Aim’s definition is: “An AI agent is a software flow controlled by a LLM, with capabilities to interact with systems and services, whether organizational or third-party.”

The common theme is an agent is a logical entity that can make decisions and take actions - whether user initiated, or on behalf of a LLM using AI techniques like reasoning, inference or pattern recognition. The other key dimension is that unlike traditional AI chatbots, agents operate with real-time contextual inference through interaction via RAG (retrieval augmented generation) systems and knowledge bases, and can autonomously perform actions based on LLM decisions.

Why do AI agents present new security & governance risks?

Building on the definition of AI agents as autonomous and operating independently, security teams face an entirely new challenge. If they want to take a deterministic approach using predefined policies and controls of what the agent can and can't do, they need to tightly scope what the agent can connect to, and which actions it can perform. However, because the motivation for adopting AI agents is both to improve productivity, as well as leverage inference, reasoning and RAG (retrieval augmented generation) for more efficient and informed decision making, security frameworks have to take into account that agents can behave in unexpected, probabilistic ways.

Agents can perform a range of autonomous actions, such as:

- Take actions on an external system, with potentially benign - but costly - overuse, unauthorized invocation of a function, or used as a vector for impersonation
- Make unauthorized or disruptive changes, rewrites to an internal database
- Perform Web crawl and leak sensitive or confidential data
- Execute code in a development or production environment - creating sandbox breakout risks

Likewise, the emergence of Model Context Protocol (MCP) to enable communication between AI applications, AI agents, applications and data sources expands the number of possible resources that agents can interact with - not least because MCP is a permissive protocol. The outcome is that AI agent security and governance is a step function increase in complexity of maintaining visibility and applying controls.

At a high level, agents present a fundamentally new set of threats because they can act autonomously:

- **Data leakage via prompt injection or unauthorized connection:** AI agents could access and share sensitive or proprietary data with internal or external users.
- **Context injection attacks:** Because of a design flaw in agentic systems, agents can be manipulated to exfiltrate sensitive, confidential or proprietary information.
- **Agent 'poisoning':** Agents can be manipulated to perform unauthorized actions, execute malicious code, exploit vulnerabilities or propagate attacks autonomously.
- **Safe and responsible use:** Because they lack context, AI agents could act on sensitive data and PII.
- **Protection of IP:** AI agents could replicate or proprietary processes, data, or content.
- **Regulatory Compliance:** Under the EU AI Act and other compliance frameworks, automated decision-making for profiling is prohibited.

UNDERSTANDING THE ATTACK SURFACE:

An initial agent taxonomy

As we mentioned, agents are not created equal. In order to implement a realistic strategy, AI governance and security teams need to take into account the specifics of what operations agents perform, the degree to which the agent is autonomous or user directed, and what resources they interact with, which data sets the agents can access, and what environment they run in.

By compiling an agent inventory and AI endpoints using a consistent taxonomy, security teams can take the next step of tracing agent activity - providing context and understanding of risky actions (what the agent is doing, and how it is executing its tasks.

Equally, enterprises will likely use a range of AI agents, and the scope for security teams to apply controls may be limited, or constrained by business considerations. For example, security teams should have the scope to enforce controls for a homegrown agent built using Langchain for a financial services application, but are unlikely to be directly involved with policies for use of a Microsoft 365 Copilot agent for a marketing use case.

Using the matrix outlined below, security, governance and enterprise architecture teams can create categories that reflect their agent use cases, and evaluate the risks of adoption.

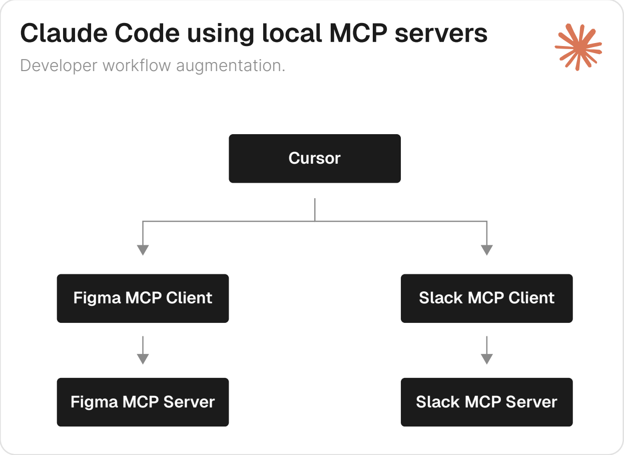
Agent Categorization Matrix

Agent Architecture	User Initiated - trusted vs untrusted	Human in the Loop for automated action approval	AI Initiated Background Agent
Tooling permissions	Internal Tooling	External Tooling (MCP Endpoints,eg)	Production Tooling
Execution Environment	SDK/Proprietary Code	Low Code	No Code
Blast Radius	Local	Self Hosted	SaaS

Using this matrix, we can add create a set of generalized agent categories:

- Local Agents: Run on the user's endpoint (e.g., Claude Code, Cursor, IDEs using MCP).
- Managed Platforms: SaaS or cloud platforms where agents are built (e.g., Copilot Studio, Google Vertex agents).
- Non-Managed/Custom Agents: Homegrown agents built in cloud environments (e.g., LangChain, Langgraph).

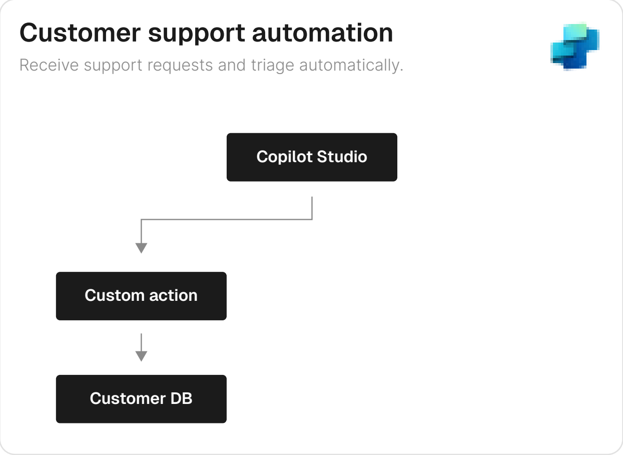
Mapping to specific agent deployment scenarios, a relatively low risk scenario would be a coding assistant initiated by a developer for code that is run using local MCP servers.



Claude Code using local MCP servers

Developer workflow augmentation.

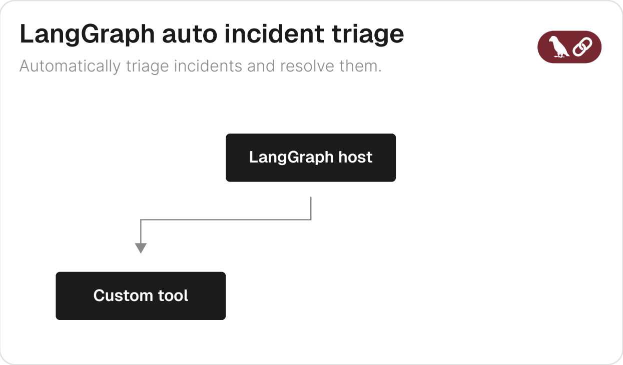
By comparison, an agent that is performing customer support automation with a human in the loop, but is taking a custom action initiated by a user prompt on customer data on shared infrastructure will have a more elevated set of risks associated.



Customer support automation

Receive support requests and triage automatically.

Further on the risk spectrum would constitute a background agent (operating without a human in the loop) performing an auto-incident triage in the LangGraph agent development environment on proprietary code in a self hosted environment, where the code execution could have unintended consequences.



LangGraph auto incident triage

Automatically triage incidents and resolve them.

Securing agents involves scoping based on:

- The profile and lineage of agent (whether a human is in the loop, whether the users associated with the agents are themselves are trusted, or the agent is relatively autonomous)
- Whether the tooling used to build and deploy the agent is trusted, is external facing or internal only
- Which datasets and infrastructure agents can connect to, and their level of authorization
- Whether guardrails and monitoring via interception point can be implemented in the environment
- Whether out of band detections and blocking are feasible if agents are acting on unsanctioned, unauthorized or untrusted data

AI Agent Security Framework and Key Principles

Discovery and Profiling

- Discovery, inventory and risk profiling of agents, agentic infrastructure (MCP servers, eg) (both for development and for production)
- Agent resource authorizations
- Agent lineage and connections
- MCP client endpoint ownership

The baseline of any security and governance program is visibility. But because of the diversity of agent types, discovery should incorporate profiling and representation of the agent's characteristics, such as degree of autonomy.

In addition, discovery and profiling should extend to MCP servers, which effectively create new endpoints for agents to interact with.

Discovery and profiling also provides value for CIOs and enterprise architecture, providing data and usage analytics on agentic adoption and trends:

Agentic Posture Management

- Insecure tooling detection - including MCP risk profiling
- Identity and data access assessment
- Toxic tool combination analysis
- Blast radius analysis - development vs production

Building on existing approaches for security posture management for custom AI applications, this principle involves mapping the agent tooling chain, as well as identity management policies applying to user initiated agents. In scope as well is supply chain risk assessment, including the agentic development environment as well as third-party MCP servers.

This security hygiene step can help security teams identify risks like third party MCP servers downloaded by developers, with known vulnerabilities or backdoors.

Observability and Monitoring

Observability in this context entails logging and auditing both how the agent acts, and what actions it performs - autonomously on behalf of a LLM, a user initiated action, or as the output of inference or reasoning.

As in the case of discovery and profiling, these types of data and insights are of value to the organization more broadly. Observability can help guide AI steering committee policies and governance frameworks.

Runtime controls

- Prompt attack detection
- Context injection detection
- Action-level role based access control scoping
- Data leakage detection

Runtime controls for agentic AI overlap with those needed for third party apps and custom built AI applications: prompt level inspection, policy enforcement for data protection and agent interaction and guardrails for output actioned by agents. In addition to enforcing role-based access control to datasets and guardrails for unsafe code execution, security teams need a mechanism to detect and block exploits that leverage the internal mechanics of LLMs.

For protection, security teams can advance two paths:

- AI detection and response using out of bound API enforcement
- AI firewall for inline enforcement

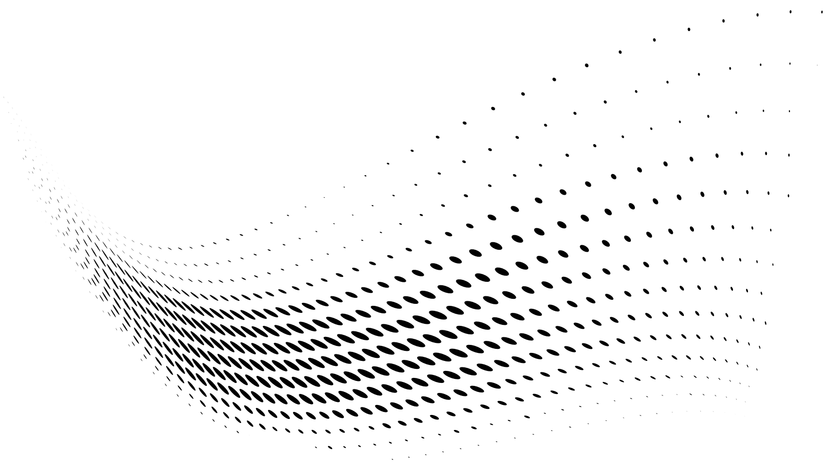
For autonomous agents, additional layers of protection, context and evaluation are required. The evaluation would involve classifying invocations as either benign or malicious, by assessing the overall context of the interaction.

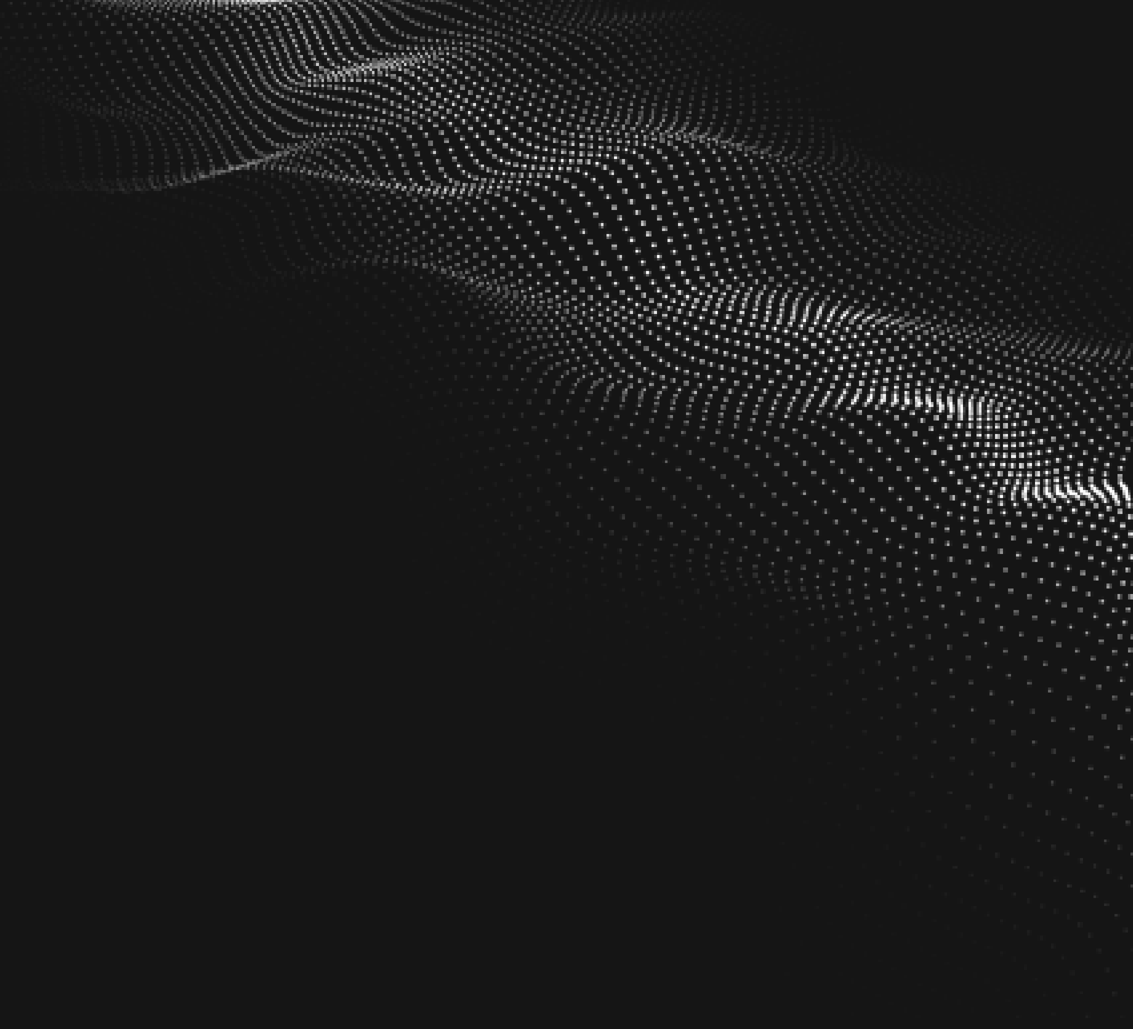
As we discussed above, the lethal trifecta of access to private data, exposure to untrusted content, and ability to communicate externally within a shared context opens the door for zero click exfiltration of data. Aim Labs’ disclosure of the EchoLeak vulnerability is a practical example of the lethal trifecta. This vulnerability is particularly challenging to address, since it is a byproduct of the benefits that agents deliver.

To detect this class of attacks, security tools must be able to detect when an agent is being attacked or misdirected and then block the action:

Setting the agenda for Agentic AI security

Security teams are responsible for providing a clear eyed view of the risk that agentic AI adoption represents. Equally, to play an integral role in harnessing the benefits of achieving business objectives, security teams must be able to present a realistic model for security and governance imperatives, with concrete and practical capabilities. As a logical outcome of the visibility and context generated by security tools, other stakeholders within the organization can also gain insights and reporting on AI agent utilization, activity, and data connections. Working with a purpose built platform underpins the role that security teams can play for the broader organization, that can be extended to new use cases as AI continues to evolve.





How Aim Security helps

Aim Security is the security partner for your AI adoption strategy. Aim Security is the one platform for all AI security needs, delivering protection, risk mitigation and governance for agents end-users adopt and home-grown AI agents. Leveraging cutting edge research from Aim Labs on emerging AI landscape threats, Aim helps organizations stay ahead of attackers.

Aim delivers a comprehensive platform for agentic AI security, including full agent discovery, inventory, and agent lineage for monitoring and tracing in tandem with flexible policy and guardrail enforcement though Aim Guards and gateway integrations.

aim security

Aim is Your Partner for the Secure AI Adoption Journey

Book a Demo