SALT

The Agentic Security Graph: A New Framework for the AI Era

Why securing the LLM layer alone leaves your enterprise exposed, and what it takes to protect the full agentic stack.

Table of Contents

01 Executive Summary

The enterprise AI stack has a blind spot. Security teams are investing heavily in LLM safeguards: prompt injection filters, output monitors, jailbreak defenses. These are necessary. But they address only one layer of a three-layer architecture.

AI agents do not just think. They act. They connect to MCP servers to access tools and workflows. They call APIs to read and write data, trigger transactions, and interact with critical business systems. This action layer, the network of MCP servers and APIs through which agents exercise real-world capabilities, is almost entirely unmonitored by today's AI security tools.

This white paper introduces the Agentic Security Graph: Salt Security's framework for understanding and securing the complete attack surface created by AI agents. It defines the three layers of agentic risk, explains why the action layer represents the most consequential and least protected attack surface in the modern enterprise, and outlines what full-stack agentic security requires.

The question is no longer whether AI agents will be deployed across your enterprise. It is whether you will be able to see them, stop them, and understand what those agents do once they are deployed.

02 The Agentic Shift: From Conversations to Consequential Actions

For the past several years, enterprise AI has been primarily a tool for generating text. Employees used LLMs to draft emails, summarize documents, and answer questions. The outputs were advisory. A human remained in the loop before anything consequential happened.

That era is ending.

Agentic AI systems are autonomous. They are designed to take action without waiting for human approval at every step. A financial services agent does not just suggest a portfolio adjustment. It executes it. A DevOps agent does not just recommend a configuration change. It applies it. A customer success agent does not just draft a refund. It processes it.

This shift from conversational AI to agentic AI is not incremental. It is categorical. The moment an AI can take action in your environment, the security calculus changes entirely.

Conversational AI	Early Agentic AI	Autonomous Agentic AI
Outputs are text. Humans decide what to do next. The blast radius of a compromised model is limited.	Agents complete discrete tasks. Some actions are automated. Human review is still common at key decision points.	Agents operate across complex workflows with minimal human intervention. Actions have immediate, real-world consequences.

The Scale of the Shift

The growth trajectory of AI agents makes this an urgent security priority, not a future one. Organizations are not planning to deploy agents. They are deploying them now. Development teams, operations teams, finance teams, and security teams themselves are all adopting agentic workflows.

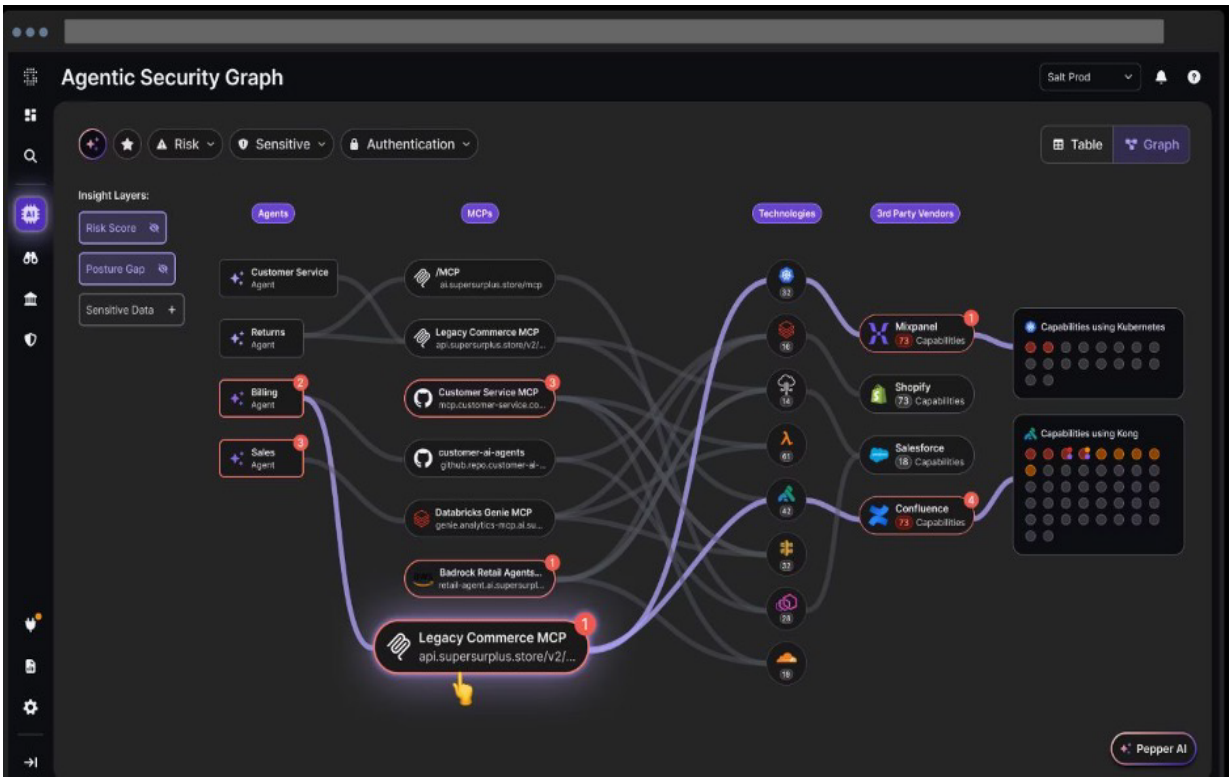
Every new agent is a new identity with capabilities, permissions, and access to enterprise systems. Every new MCP server is a new integration point connecting agents to tools and data. Every new API call is a new action taken on behalf of an AI system that can be manipulated, misdirected, or compromised.

03 The Agentic Security Graph: Three Layers, One Attack Surface

Most AI security frameworks focus on the model. They treat the LLM as the primary risk surface and invest in guardrails around what goes in and what comes out. This framing made sense when AI was conversational. It is dangerously incomplete when AI is agentic.

An AI agent operates across three distinct layers. Each layer has its own risk profile. Each requires its own security controls. And all three are connected, meaning a compromise at any layer can propagate through the others.

This interconnected structure is what Salt Security calls the Agentic Security Graph.



Layer 1: The LLM (The Brain)

The large language model is the reasoning engine of an agentic system. It interprets instructions, plans sequences of actions, and determines what to do next. The LLM layer is where decisions are made.

Threats at this layer are well-documented. Prompt injection attacks attempt to hijack the model's decision-making by inserting malicious instructions into the input stream. Jailbreaks attempt to bypass safety training. Model poisoning can introduce biased or backdoored behaviors during training or fine-tuning.

These are real threats, and the security industry has developed meaningful tools to address them. But they address only the decision layer. An agent that makes the right decision can still be exploited at the layer where it acts on that decision.

Layer 2: MCP Servers (The Hands)

Model Context Protocol (MCP) servers are the connective tissue of the agentic stack. They expose tools, APIs, data sources, and business logic to AI agents in a standardized way. When an agent needs to send a Slack message, query a database, trigger a workflow, or call an external service, it does so through an MCP server.

MCP is emerging as the dominant standard for agent-to-tool connectivity. Its rapid adoption means that within a short window, most enterprise AI agents will be interacting with systems through MCP servers.

This layer is almost entirely unmonitored today. Security teams lack visibility into what MCP servers exist in their environment, what tools they expose, what permissions they carry, and how agents are using them. A compromised or misconfigured MCP server is a direct path to enterprise systems, and most organizations would have no way to detect it or tools to address them. An agent that makes the right decision can still be exploited at the layer where it acts on that decision.

MCP servers are the fastest-growing and least-secured integration point in the enterprise AI stack. Most organizations do not have a complete inventory of the MCP servers operating in their environment right now.

Layer 3: APIs (The Action Layer)

APIs are where agents touch the real world. Every read, write, update, deletion, transaction, and workflow trigger that an agent initiates flows through an API. The API layer is the ultimate execution surface of agentic AI.

Salt Security has spent eight years securing APIs. That experience revealed something important: APIs are extraordinarily difficult to secure because they are diverse, dynamic, and deeply integrated into every system an organization operates. The attack surface is large, the traffic is high-volume, and the behaviors are complex.

When AI agents enter the picture, API security becomes exponentially more complex. Agents generate API traffic that is difficult to distinguish from legitimate traffic, but that can carry instructions shaped by manipulation at the LLM or MCP layer. They can be directed to enumerate resources, exfiltrate data, escalate privileges, or take destructive actions. All of this flows through API calls that individually may look unremarkable.

Without the ability to understand the full context of an API call (where it originated, what agent made it, what MCP server mediated it, what user or workflow authorized it) security teams cannot assess whether any given API transaction is legitimate or malicious.

04 The Security Industry's Blind Spot

The enterprise AI stack has a blind spot. Security teams are investing heavily in LLM safeguards: prompt injection filters, output monitors, jailbreak defenses. These are necessary. But they address only one layer of a three-layer architecture.

Tool Category	What It Covers and What It Misses
Prompt Injection Defense	Monitors inputs to the LLM to detect attempts to hijack agent behavior. Does not monitor what the agent does once it decides to act.
Output Filtering	Inspects LLM outputs for harmful or sensitive content. Does not track what happens when that output triggers downstream API calls.
Model Observability	Provides visibility into LLM performance, token usage, and output quality. Not designed to detect API abuse or MCP server anomalies.
API Gateways	Enforce rate limits and authentication. Lack the behavioral intelligence to detect AI-driven attacks or agent-specific abuse patterns.

AI agents do not just think. They act. They connect to MCP servers to access tools and workflows. They call APIs to read and write data, trigger transactions, and interact with critical business systems. This action layer, the network of MCP servers and APIs through which agents exercise real-world capabilities, is almost entirely unmonitored by today's AI security tools.

The Agentic Security Graph is Salt Security's framework for understanding and securing the complete attack surface created by AI agents. It defines the three layers of agentic risk, explains why the action layer represents the most consequential and least protected attack surface in the modern enterprise, and outlines what full-stack agentic security requires.

The result of this gap is an enterprise that believes it is protecting its AI investment but is leaving its most consequential exposure unaddressed. The model may be locked down. The action layer is wide open.

05 Why Salt Security Is Uniquely Positioned

Salt Security did not randomly move to agentic security. We arrived here through eight years of building the industry's most advanced API security platform.

API security at enterprise scale requires something that most security tools lack: the ability to understand behavior across millions of API actions, build a baseline of what normal looks like, and detect anomalies that indicate attack. This requires AI-driven analytics, deep protocol understanding, and a data asset built from years of observing real-world API traffic across hundreds of enterprise deployments.

That foundation is exactly what agentic security demands.

Eight Years of API Intelligence

More API behavioral data than any other vendor. The models that detect agentic attack patterns are trained on real-world enterprise traffic, not synthetic datasets.

MCP Security Expertise

Salt was among the first security vendors to build purpose-built controls for MCP servers, the emerging action layer connecting agents to enterprise tools and workflows.

Full-Stack Architecture

The only platform that discovers, governs, and protects all three layers of the Agentic Security Graph: LLM, MCP servers, and APIs, from development through runtime.

The Salt Agentic Security Platform

Salt's Agentic Security Platform operationalizes the Agentic Security Graph framework through two core capability sets.

AG-SPM: Agentic Security Posture Management
AG-SPM gives security teams continuous visibility into the full Agentic Security Graph. It discovers all AI agents operating in the environment, maps the MCP servers they connect to, inventories the APIs they call, and assesses the permissions and configurations across all three layers. When posture drifts (a new MCP server appears, an agent accumulates excessive privileges, an API endpoint is exposed without adequate controls) AG-SPM detects it and enables remediation before attackers can exploit it.

AG-DR: Agentic Detection and Response
AG-DR provides real-time threat detection and response across the Agentic Security Graph. It monitors API traffic generated by AI agents, applies behavioral analytics to identify patterns that indicate compromise or abuse, and enables security teams to investigate and respond to threats that cross layer boundaries. Because AG-DR understands the full context of agentic activity rather than individual API calls in isolation, it can connect signals across layers to identify sophisticated attacks that would be invisible to point solutions.

06 The Agentic Threat Landscape

The threats facing agentic systems are not theoretical. They are already being observed in production environments. Understanding the attack surface of the Agentic Security Graph requires understanding how adversaries are beginning to exploit it.

Prompt Injection via the Action Layer

Prompt injection attacks traditionally target the LLM input directly. But agents often retrieve content from external sources as part of their task execution: web pages, documents, database records, emails. Any of this content can contain injected instructions designed to redirect agent behavior. When the agent acts on those instructions, it does so through MCP servers and APIs, giving attackers a path to enterprise systems that bypasses the LLM guardrails entirely.

MCP Server Compromise and Misconfiguration

An MCP server with excessive permissions is a vulnerability waiting to be exploited. A compromised MCP server can redirect agents to malicious endpoints, exfiltrate credentials, or silently modify the tools an agent believes it is using. Because most organizations lack complete visibility into their MCP server inventory, these misconfigurations often go undetected until a breach occurs.

The most dangerous agentic attacks will not look like attacks at the layer where they originate. They will exploit gaps between layers, where no single security tool has visibility.

API Abuse Through Legitimate Agent Credentials

AI agents are provisioned with service accounts and API credentials that enable them to perform their designated tasks. An agent whose behavior is manipulated at the LLM or MCP layer can use those legitimate credentials to make API calls that exfiltrate data, modify records, or trigger high-impact workflows. None of this triggers authentication-based controls because the credentials are valid. Only behavioral analytics that understands what normal agent activity looks like can detect this class of attack.

Agent-to-Agent Attack Propagation

Complex agentic systems increasingly involve multiple agents working in concert. An orchestrator agent may delegate tasks to specialized subagents. This multi-agent architecture creates a new propagation path for attacks: compromise one agent, and it can inject malicious instructions into agents it coordinates with. Security controls that focus on individual agents cannot detect attacks that move through the collaboration layer.

07 The Agentic Security Graph in Practice

The Agentic Security Graph is not just a conceptual framework. It is an operational guide for how security teams should approach the AI era.

Step 1: Discover the Full Graph

You cannot secure what you cannot see. The first step is achieving complete visibility into every agent, every MCP server, and every API in your environment. This is harder than it sounds. Agents are deployed by development teams, business units, and individual contributors. MCP servers emerge organically as teams adopt new AI tools. APIs evolve constantly.

Salt's discovery capability continuously maps the Agentic Security Graph across cloud environments, development pipelines, and runtime traffic, giving security teams the complete inventory they need to understand their actual attack surface.

Step 2: Govern Posture Across All Three Layers

Discovery is the foundation. Governance is the ongoing discipline. Security teams need to define what good looks like across the Agentic Security Graph: what permissions agents should carry, what MCP servers they should access, what API behaviors are within scope.

AG-SPM enables policy-based governance at scale, surfacing deviations from defined posture, prioritizing remediation by risk, and providing the documentation that compliance and audit processes require.

Step 3: Detect and Respond in Real Time

Posture management addresses known risks. AG-DR addresses active threats. By combining deep API behavioral analytics with context from the full Agentic Security Graph, Salt can detect attacks that cross layer boundaries, identifying the difference between an agent performing its intended function and an agent that has been manipulated into doing something it should not.

When a threat is detected, the response workflow needs to be fast and precise. Salt provides the context security teams need to understand the scope of an incident, contain it, and remediate the underlying exposure.

08 Conclusion: The Action Layer Cannot Wait

The enterprise AI transformation is already underway. Agents are being deployed, workflows are being automated, and the action layer is expanding every day. Security teams that wait for the threat to materialize before building agentic security controls will find themselves responding to breaches rather than preventing them.

The Agentic Security Graph provides the framework to act now. It defines the full attack surface. It identifies the gaps in current coverage. And it maps a path to complete protection across the LLM, MCP, and API layers that together determine what AI agents can do in your environment.

Salt Security brings eight years of API security leadership to this challenge. That is not a credential. It is a requirement. Securing the action layer demands exactly the capabilities Salt has spent nearly a decade building.

The question every CISO needs to answer is not whether to secure agentic AI. It is how fast that security can be put in place.

The most dangerous agentic attacks will not look like attacks at the layer where they originate. They will exploit gaps between layers, where no single security tool has visibility.

09

About Salt Security

Salt Security is the global leader in agentic security. Founded in 2018, Salt pioneered the API security category and has spent eight years building the industry's most advanced behavioral analytics platform. Today, Salt protects the full Agentic Security Graph, securing the LLMs, MCP servers, and APIs that modern AI agents depend on. Hundreds of enterprises across financial services, healthcare, technology, and retail trust Salt to secure their most critical interfaces.

Take the Next Step:

Do not rely on inadequate tools to protect your API and AI infrastructure.
See the Salt difference for yourself.

Get a demo