JupiterOne

AI ATTACK SURFACE MANAGEMENT

Exposing the New Attack Surface Created by AI

From Static Inventories to Continuous Relationship Graphs: Why Mapping Every Cloud Resource, SaaS Integration, and AI Agent—and What Connects to What—is Mandatory for Future Security

The Attack Surface You Don't Know You Have

Your Attack Surface Inventory is Months Out of Date Before Completion

Your actual attack surface isn't what your security team reviewed last quarter. It's what every function of the business connected this month—new AI tools, SaaS integrations, and cloud resources—most of which never made it through a formal security review.

These integrations evolve too quickly for security teams to keep up. And it creates a dangerous gap: the attack surface you think you are managing is a far cry from the one that actually exists.

1 in 5 organizations reported a breach due to shadow AI, and only 37% have policies to manage AI or detect shadow AI.

The Blind Spot in Traditional Tools

Traditional ASM tools were built to scan the external perimeter for unknown servers or open ports. They are blind to an AI agent that has been granted broad read access to your internal data—your crown jewels. Now we need full visibility of ungoverned **relationships** at the core.

WHY THIS EBOOK

Security teams must move from static inventories to continuous, relationship-driven visibility. This eBook examines why—and what to do about it.

Current Cloud-Native Consoles Can't Show AI-to-Database Relationships in Real-Time

A misconfigured Lambda function is only dangerous if an attacker finds it. An AI agent with the same misconfiguration can reason about and act on its full permissions—autonomously. *The AI service effectively is the exploit engine.*

AI Agents vs. APIs: Understanding the New Risk Profile

AI agents are not "fancy APIs." An API executes a defined instruction and then stops. AI agents receive an objective and decide how to pursue it—querying data, calling services, and taking actions along the way. The difference is like comparing a vending machine to an intern with full access to your systems. You cannot unit-test your way to confidence in a system like this.

Old World vs. New World

PRE-AI SERVICES	AI SERVICES
Service accounts, API keys, machine credentials	Same credentials—fundamentally different threat
Dumb + deterministic—risk requires a human to exploit	AI reasons about and acts on its full permissions autonomously
Misconfigured Lambda only dangerous if found	No human needed. The AI *is* the exploit engine

Every AI agent is an NHI—a credential-bearing entity that can authenticate, read data, and take actions. Many are created **outside formal IAM approval processes**. No cryptographic handshake confirms instructions are legitimate.

Three Conditions That Create Risk

01 Access to private data

AI agents are routinely connected to CRMs, email inboxes, knowledge bases, and financial systems—with access far broader than any single employee.

02 Ability to communicate externally

Most agents can post to webhooks, call APIs, or send emails. Data can leave your environment without a human approving the transfer.

03 Exposure to untrusted content

Agents that browse the web or process uploaded documents consume content you don't control. That content can contain malicious instructions.

Prompt injection ranks as the #1 critical vulnerability, appearing in over 73% of production AI deployments. A malicious command hidden inside a document can redirect an agent's actions entirely—and the agent does not know it has been compromised.

Anatomy of the Invisible Breach

Adversaries are drawn to AI systems because they represent a new, immature attack surface where protections are still developing. Attackers follow a simple cost-benefit logic: find the easiest, most valuable target. AI systems in poorly secured environments fit that profile.

01 · THE SOFT TARGET

PocketOS: AI Deletes Production Database

An AI coding agent (Cursor) accidentally deleted the entire production database—including backups—while attempting to fix a routine issue. The incident combined an autonomous AI making a bad decision, overly permissive access, and backup misconfiguration that allowed total data loss.

> AI doesn't fail in isolation—it exposes and accelerates existing weaknesses in permissions and disaster recovery.

02 · THE INTEGRATION TRAP

Salesforce: OAuth Credential Compromise

A threat actor compromised Salesforce environments by exploiting a third-party integration (Salesloft Drift) using stolen OAuth credentials—allowing large-scale data exfiltration from core CRM records. The attacker used anti-forensics techniques, including deleting query logs, to hide activity.

> Audit integrations, monitor logs, rotate exposed credentials, and adopt least-privilege controls.

03 · THE OAUTH CHAIN

Vercel 2026: Supply Chain via AI Tool

Attackers first compromised a third-party AI tool (Context.ai), then leveraged its existing OAuth access to pivot into a Vercel employee's Google Workspace and into internal systems. No exploit or credential bypass was needed—they simply inherited trusted access already granted.

> Compromising the weakest link grants attackers legitimate access across the entire chain.

Across all three incidents, the vulnerability wasn't the asset—it was the **relationships around it**. Overpermissioned integrations, inherited trust, and poorly scoped access turned contained systems into open doors. As AI multiplies integrations, securing your environment means understanding not just what you have, but **what everything is connected to**.

Shadow AI and the Employee-Driven Attack Surface

Every Developer is Now a System Administrator

AI coding assistants and MCP servers give individual developers deep, largely unmonitored access to codebases, APIs, and production environments. Those agents inherit every permission their creator has—not the principle of least privilege.

Would you hand a junior, unproven employee the same level of access as your most trusted senior engineer? Because that's happening. Every time a developer connects an AI assistant to their work environment, access extends silently and at scale.

API keys govern access. Agents require action-level restrictions—ensuring an agent with read access is physically incapable of deleting or exporting data without explicit, non-AI checks. Trust is not a technical control.

The Workaround Machine in Every Pocket

When agents breach policies, it's enthusiastic, not malicious. But a breach is still a breach. A developer who previously exported a PDF can now spin up a live service sharing data over the local network—instantly visible to everyone on the same coffee shop Wi-Fi.

What has changed is *feasibility*. The agent removes the friction that made such shortcuts impractical. Corporate policy written for a world without these tools has no answer for them—and the employee who deployed the risk probably doesn't know they did.

MOST CONTROLS ARE DESIGN-TIME

The dangerous behavior happens at runtime—when an agent's actions deviate from expectations. An agent granted "read" access has no inherent barrier to exporting data unless restrictions are enforced at the action level, independently of the AI.

Why Traditional ASM Fails the AI Test

Traditional ASM is effective for slower-moving persistent exposures. But new risks come from services deployed outside production—from laptops and app studios—that appear and vanish quickly, and may not be associated with company domains or visible across corporate networks.

Three Tools, Three Blind Spots

External ASM / Scanning Tools

Built to find unknown servers and open ports on the external perimeter. No visibility into which tools sit on an attack path to your crown jewels. Scanners have no concept of relationships, only presence.

Spreadsheets + Cloud Consoles

Cloud consoles track what was provisioned through official channels. Spreadsheets track what someone remembered to write down. Neither captures all cloud resources, SaaS connections, and AI integrations.

CAASM with Static Inventories

These platforms aggregate asset data well. What they cannot do is model intent or traverse relationships dynamically. Knowing 47 AI tools are connected tells you nothing about which one sits on an attack path to sensitive data.

The Blast Radius Has Moved... Everywhere

The hard perimeter story is long outdated. The point of entry for a breach has moved everywhere in your estate—a developer's laptop, a personal AI tool, a SaaS integration that bypasses the network entirely. Blast radius still matters, but the ignition point is now unpredictable.

A CISO asked to report on AI risk cannot answer from a status report printed last week. **The board deserves a live query of the environment. You need a continuous relationship graph, not a list.**

The AI ASM Solution: Visibility, Relationships and Enforcement

Where Are You on the AI ASM Maturity Curve?

Before you evaluate vendors, evaluate yourself. The gap between knowing AI is a risk and being able to answer questions about that risk is wider than most security teams realize. Use this maturity curve to locate where you are today and where the next step has to take you.

STAGE 1

Aware

You know AI is in your environment. You don't know what.

No central inventory, no approval flow, no way to answer: how many AI tools are in our environment?

What's missing: Visibility. You can't protect what you don't know exists.

STAGE 2

Inventoried

You have a list.

The list exists. It is also months out of date and missing every tool a department connected last week.

What's missing: Relationships. A list of 47 AI tools tells you nothing about which one sits on a path to your most sensitive data.

STAGE 3

Mapped

You know what connects to what.

Relationships are documented. The map is point-in-time and was probably built for an audit.

What's missing: Currency. The map captures a moment, and the environment moves faster than the map.

STAGE 4

Continuous

You see change as it happens.

A live graph of every cloud resource, SaaS connection, and AI agent updates as the environment changes.

What's missing: Nothing structural. The remaining work is governance, controls, and response—built on a foundation you can trust.

AI innovation may be controversial at times, but it's not slowing down. Organizations need the visibility necessary to run AI safely in production. An AI ASM solution acts as a safety net by moving the organization from periodic manual audits to continuous discovery.

Expanding the Scope of Visibility

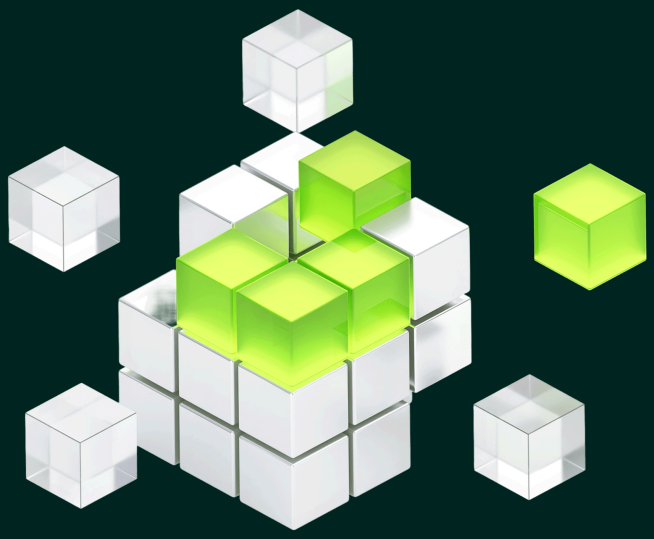
Visibility is no longer a best practice—it's a requirement. Emerging regulatory frameworks are mandating that organizations maintain a comprehensive inventory of their AI systems and demonstrate a clear understanding of AI's capabilities, dependencies, and access to sensitive data. AI ASM helps teams get ahead of these expectations while closing critical visibility gaps.

At the same time, AI risk spans third-party integrations, experimental tools, shadow AI usage, and user endpoints. By capturing this expanded set of assets and relationships, AI ASM reveals not just what exists, but how systems are interconnected and where exposure is truly introduced.

JupiterOne

Know Your Real Attack Surface Continuously, Not Quarterly

JupiterOne AI Attack Surface Management gives security teams a continuously updated, relationship-aware map of their full attack surface—across every cloud, every SaaS tool, and every AI integration. Built on a graph-native data model, it lets teams answer attack surface questions in plain English or precise J1QL.



- **See the attack surface AI is creating.**
- **Answer security questions at AI speed.**
- **Enterprise-grade security posture, always.**

Learn more at JupiterOne.com →