Kiteworks

SOLUTION BRIEF

Six AI Vulnerabilities. Three Gaps. One Architectural Answer.

A compliance analyst reads a vulnerability advisory. Salesforce just patched ForcedLeak — a prompt injection hidden in a Web-to-Lead form field that caused an AI agent to query the CRM and exfiltrate records to a domain the attacker bought for five dollars. She checks her firm's Agentforce deployment. The AI touches hundreds of leads per day with access to the full CRM. She asks security whether any control — independent of the AI model — evaluates each data request against policy before the AI retrieves it.

The answer is no. That gap cannot be closed with a patch.

Six Disclosures. Three Gaps. One Year.

Between mid-2025 and April 2026, researchers disclosed six critical AI vulnerabilities across the platforms enterprises rely on daily. Every vendor patched. Every attack exploited architectural gaps that no patch closes.

Vulnerability	Platform	Gap(s)	What Failed
EchoLeak (CVSS 9.3)	Microsoft 365 Copilot	**1 + 2**	Crafted email as AI context; broad implicit access; exfiltration via trusted domains
ForcedLeak (CVSS 9.4)	Salesforce Agentforce	**1 + 2**	Form field as AI context; full CRM access; $5 expired allowlisted domain
GeminiJack	Google Gemini Enterprise	**1 + 2**	Poisoned doc indexed by RAG; zero-click sweep across all Workspace data
Reprompt	Microsoft Copilot	**1 + 2**	URL parameter as AI context; single-click exfiltration
GrafanaGhost	Grafana AI	**1 + 3**	Event log data as AI context; system-level process with excessive functional scope. RBAC exists — never triggered
OpenAI Plugin	OpenAI Ecosystem	**2 + 3**	Compromised plugin harvested credentials; 6 months, 47 enterprises

These six contain three distinct gaps:

Gap 1: Untrusted input as trusted AI context. Every attack begins with external data — an email, a shared document, a form field, event log entries — entering the system and being processed by AI without validation. Present in all six. The industry is largely ignoring it.

Gap 2: Broad AI data access without per-operation enforcement. Five of six involve AI that authenticated once and then accessed everything in scope. No per-operation policy evaluation constrained each retrieval.

Gap 3: Process containment failures. GrafanaGhost operated through system-level back-end processes — not a user session. The process had functional scope it was never designed to use. The OpenAI plugin attack succeeded because credentials were stored where compromised code could access them.

Model-level guardrails failed in every case. Grafana's fell to a single keyword. Salesforce's CSP was bypassed for five dollars. Guardrails are configuration settings inside the system being attacked. They supplement real controls. They substitute for none of them.

The Architectural Answer

Gap 2 — the data access gap that five of six vulnerabilities exploited — has a direct architectural answer. Kiteworks Compliant AI sits between AI agents and regulated data, enforcing governance at the data layer where the AI model cannot bypass it. Every interaction passes through four checkpoints, independent of the model, prompt, or agent framework:

Authenticated Identity. Every agent verified via OAuth 2.0, linked to the human authorizer. Credentials stored in the OS keychain — never exposed to the AI model. The control the OpenAI plugin attack was missing.

Policy-Enforced Access. Every request evaluated in real time against agent identity, data classification, and context using ABAC. Minimum necessary access enforced at the operation level — not the session level. The control EchoLeak, GeminiJack, and ForcedLeak were missing.

FIPS 140-3 Validated Encryption. All agent-accessed data encrypted in transit and at rest with validated cryptographic modules.

Tamper-Evident Audit Trail. Every interaction logged with full attribution and streamed to SIEM in real time. The evidence that regulators require — and that the six vulnerabilities left no trace of.

The architectural principles that make these controls effective — independence from the AI model, enforcement below the prompt layer, credential isolation outside the AI's context — extend to gaps one and three. Kiteworks' MCP implementation enforces per-operation policy, validates path traversal, and keeps credentials outside the model's reach. The framework is built to extend as the threat evolves.

Three Questions That Tell You Where You Stand

Gap 1 (Input Trust): Do you validate externally sourced data before AI processes it? If not, every vulnerability in this series would succeed in your environment.

Gap 2 (Data Access): Does every AI data request get evaluated against policy independently — on every operation, not at the session level? If not, one injected instruction can sweep your entire data scope.

Gap 3 (Containment): Do you know which APIs and output channels your back-end AI processes can invoke? If not, you have assessed data access but not functional scope. Combined with untrusted input, that is GrafanaGhost.

Learn More

Kiteworks