Sechs KI-Schwachstellen. Drei Lücken. Eine architektonische Antwort.

Eine Compliance-Analystin liest eine Schwachstellenmeldung. Salesforce hat gerade ForcedLeak gepatcht – eine Prompt-Injection, versteckt in einem Web-to-Lead-Formularfeld, die dazu führte, dass ein KI-Agent das CRM abfragte und Datensätze an eine vom Angreifer für fünf Dollar gekaufte Domain exfiltrierte. Sie prüft die Agentforce-Implementierung ihres Unternehmens. Die KI verarbeitet täglich Hunderte von Leads mit Zugriff auf das gesamte CRM. Sie fragt die Sicherheitsabteilung, ob eine Kontrolle – unabhängig vom KI-Modell – jede Datenanfrage vor dem Zugriff der KI anhand von Richtlinien prüft.

Die Antwort lautet nein. Diese Lücke lässt sich nicht mit einem Patch schließen.

Sechs Offenlegungen. Drei Lücken. Ein Jahr.

Zwischen Mitte 2025 und April 2026 haben Forscher sechs kritische KI-Schwachstellen auf Plattformen offengelegt, auf die Unternehmen täglich angewiesen sind. Jeder Anbieter hat gepatcht. Jeder Angriff nutzte architektonische Lücken aus, die kein Patch schließen kann.

Schwachstelle	Plattform	Lücke(n)	Was fehlgeschlagen ist
EchoLeak (CVSS 9.3)	Microsoft 365 Copilot	1 + 2	Manipulierte E-Mail als KI-Kontext; weitreichender impliziter Zugriff; Exfiltration über vertrauenswürdige Domains
ForcedLeak (CVSS 9.4)	Salesforce Agentforce	1 + 2	Formularfeld als KI-Kontext; voller CRM-Zugriff; abgelaufene, für fünf Dollar erworbene Allowlist-Domain
GeminiJack	Google Gemini Enterprise	1 + 2	Manipuliertes Dokument, indexiert durch RAG; Zero-Click-Sweep über alle Workspace-Daten
Reprompt	Microsoft Copilot	1 + 2	URL-Parameter als KI-Kontext; Exfiltration mit nur einem Klick
GrafanaGhost	Grafana AI	1 + 3	Event-Log-Daten als KI-Kontext; Systemprozess mit übermäßigem Funktionsumfang. RBAC existiert – wurde nie ausgelöst
OpenAI Plugin	OpenAI Ecosystem	2 + 3	Kompromittiertes Plugin sammelte Zugangsdaten; 6 Monate, 47 Unternehmen

Diese sechs Schwachstellen offenbaren drei zentrale Lücken:

Lücke 1: Ungeprüfte Eingaben als vertrauenswürdiger KI-Kontext. Jeder Angriff beginnt mit externen Daten – einer E-Mail, einem geteilten Dokument, einem Formularfeld, Event-Log-Einträgen –, die ins System gelangen und von der KI ohne Validierung verarbeitet werden. In allen sechs Fällen vorhanden. Die Branche ignoriert dieses Problem weitgehend.

Lücke 2: Umfassender KI-Datenzugriff ohne durchgängige Richtlinienprüfung. In fünf von sechs Fällen authentifizierte sich die KI einmal und griff dann auf alle Daten im Geltungsbereich zu. Keine Richtlinienprüfung auf Vorgangsebene schränkte die einzelnen Abrufe ein.

Lücke 3: Fehler bei der Prozessbegrenzung. GrafanaGhost agierte über systemweite Backend-Prozesse – nicht über eine Anwendersitzung. Der Prozess hatte einen Funktionsumfang, für den er nie vorgesehen war. Der Angriff auf das OpenAI-Plugin war erfolgreich, weil Zugangsdaten dort gespeichert waren, wo kompromittierter Code darauf zugreifen konnte.

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.

Die architektonische Antwort

Lücke 2 – der Datenzugriffs-Gap, den fünf der sechs Schwachstellen ausnutzten – hat eine direkte architektonische Lösung. Kiteworks Compliant AI sitzt zwischen KI-Agenten und regulierten Daten und erzwingt Governance auf der Datenebene, die das KI-Modell nicht umgehen kann. Jede Interaktion durchläuft vier Kontrollpunkte, unabhängig vom Modell, Prompt oder Agent-Framework:

Authentifizierte Identität. Jeder Agent wird über OAuth 2.0 verifiziert und mit dem menschlichen Autorisierer verknüpft. Zugangsdaten werden im OS-Schlüsselbund gespeichert – nie im KI-Modell offengelegt. Die Kontrolle, die beim OpenAI-Plugin-Angriff fehlte.

Richtlinienbasierter Zugriff. Jede Anfrage wird in Echtzeit anhand der Agentenidentität, Datenklassifizierung und des Kontexts mittels ABAC geprüft. Minimal notwendiger Zugriff wird auf Vorgangsebene durchgesetzt – nicht auf Sitzungsebene. Die Kontrolle, die bei EchoLeak, GeminiJack und ForcedLeak fehlte.

FIPS 140-3-validierte Verschlüsselung. Alle von Agenten abgerufenen Daten werden während der Übertragung und im ruhenden Zustand mit validierten kryptografischen Modulen verschlüsselt.

Manipulationssicherer Audit-Trail. Jede Interaktion wird mit vollständiger Zuordnung protokolliert und in Echtzeit an SIEM gestreamt. Der Nachweis, den Aufsichtsbehörden verlangen – und den die sechs Schwachstellen nicht hinterließen.

Die architektonischen Prinzipien, die diese Kontrollen wirksam machen – Unabhängigkeit vom KI-Modell, Durchsetzung unterhalb der Prompt-Ebene, Isolierung von Zugangsdaten außerhalb des KI-Kontexts – lassen sich auch auf die Lücken eins und drei übertragen. Die MCP-Implementierung von Kiteworks erzwingt Richtlinien auf Vorgangsebene, validiert Pfad-Traversierungen und hält Zugangsdaten außerhalb der Reichweite des Modells. Das Framework ist darauf ausgelegt, sich mit der Bedrohungslage weiterzuentwickeln.

Drei Fragen, die zeigen, wo Sie stehen

Lücke 1 (Input Trust): Validieren Sie externe Datenquellen, bevor die KI sie verarbeitet? Wenn nicht, würde jede Schwachstelle dieser Serie in Ihrer Umgebung funktionieren.

Lücke 2 (Data Access): Wird jede KI-Datenanfrage unabhängig – bei jedem Vorgang, nicht nur auf Sitzungsebene – gegen Richtlinien geprüft? Wenn nicht, kann eine injizierte Anweisung Ihren gesamten Datenbestand kompromittieren.

Lücke 3 (Containment): Wissen Sie, welche APIs und Ausgabekanäle Ihre Backend-KI-Prozesse aufrufen können? Wenn nicht, haben Sie den Datenzugriff bewertet, aber nicht den Funktionsumfang. In Kombination mit ungeprüften Eingaben entsteht daraus GrafanaGhost.

Erfahren Sie mehr: