Six vulnérabilités de l'IA. Trois failles. Une réponse architecturale.

Une analyste conformité lit un avis de vulnérabilité. Salesforce vient de corriger ForcedLeak — une injection de prompt cachée dans un champ de formulaire Web-to-Lead qui a permis à un agent IA d'interroger le CRM et d'exfiltrer des enregistrements vers un domaine acheté cinq dollars par l'attaquant. Elle vérifie le déploiement Agentforce de son entreprise. L'IA traite des centaines de leads par jour avec un accès complet au CRM. Elle demande à l'équipe sécurité si un contrôle — indépendant du modèle IA — évalue chaque demande de données par rapport à la politique avant que l'IA n'y accède.

La réponse est non. Cette faille ne se corrige pas par un simple patch.

Six divulgations. Trois failles. Un an.

Entre mi-2025 et avril 2026, des chercheurs ont révélé six vulnérabilités critiques de l'IA sur les plateformes utilisées quotidiennement par les entreprises. Tous les éditeurs ont publié des correctifs. Toutes les attaques ont exploité des failles architecturales impossibles à corriger par un patch.

Vulnérabilité	Plateforme	Faille(s)	Échec constaté
EchoLeak (CVSS 9.3)	Microsoft 365 Copilot	1 + 2	Email conçu comme contexte IA ; accès implicite étendu ; exfiltration via domaines de confiance
ForcedLeak (CVSS 9.4)	Salesforce Agentforce	1 + 2	Champ de formulaire comme contexte IA ; accès complet au CRM ; domaine autorisé expiré à 5 $
GeminiJack	Google Gemini Enterprise	1 + 2	Document piégé indexé par RAG ; balayage sans clic sur toutes les données Workspace
Reprompt	Microsoft Copilot	1 + 2	Paramètre d'URL comme contexte IA ; exfiltration en un clic
GrafanaGhost	Grafana AI	1 + 3	Données de logs comme contexte IA ; processus système avec périmètre fonctionnel excessif. RBAC présent — jamais déclenché
OpenAI Plugin	OpenAI Ecosystem	2 + 3	Plugin compromis ayant collecté des identifiants ; 6 mois, 47 entreprises

Ces six attaques révèlent trois failles distinctes :

Faille 1 : Entrée non fiable utilisée comme contexte IA de confiance. Chaque attaque commence par des données externes — email, document partagé, champ de formulaire, logs d'événements — intégrées au système et traitées par l'IA sans validation. Présente dans les six cas. Le secteur l'ignore largement.

Faille 2 : Accès étendu aux données par l'IA sans contrôle opérationnel. Cinq sur six impliquent une IA authentifiée une fois, puis accédant à toutes les données dans son périmètre. Aucun contrôle de politique par opération ne limite chaque extraction.

Faille 3 : Défaut de confinement des processus. GrafanaGhost a opéré via des processus back-end système – hors session utilisateur. Le processus disposait de droits fonctionnels non prévus. L'attaque via le plugin OpenAI a réussi car les identifiants étaient stockés à un endroit accessible par du code compromis.

Les garde-fous au niveau du modèle ont échoué à chaque fois. Celui de Grafana a cédé à un simple mot-clé. La CSP de Salesforce a été contournée pour cinq dollars. Les garde-fous sont des réglages internes au système attaqué. Ils complètent les vrais contrôles, mais ne les remplacent jamais.

La réponse architecturale

La faille 2 — l'accès aux données exploité par cinq des six vulnérabilités — appelle une réponse architecturale directe. Kiteworks Compliant AI s'intercale entre les agents IA et les données réglementées, en appliquant la gouvernance au niveau des données, là où le modèle IA ne peut pas la contourner. Chaque interaction passe par quatre points de contrôle, indépendants du modèle, du prompt ou du framework agent :

Identité authentifiée. Chaque agent est vérifié via OAuth 2.0, lié à l'utilisateur autorisant. Les identifiants sont stockés dans le trousseau OS — jamais exposés au modèle IA. C'est le contrôle qui manquait lors de l'attaque du plugin OpenAI.

Accès contrôlé par politique. Chaque requête est évaluée en temps réel selon l'identité de l'agent, la classification des données et le contexte via ABAC. L'accès minimal nécessaire est appliqué à l'opération — pas à la session. C'est le contrôle qui faisait défaut pour EchoLeak, GeminiJack et ForcedLeak.

Chiffrement validé FIPS 140-3. Toutes les données consultées par l'agent sont chiffrées en transit et au repos avec des modules cryptographiques validés.

Traçabilité infalsifiable. Chaque interaction est journalisée avec attribution complète et transmise en temps réel au SIEM. C'est la preuve exigée par les régulateurs — et que les six vulnérabilités n'ont jamais laissée.

Les principes architecturaux qui rendent ces contrôles efficaces — indépendance vis-à-vis du modèle IA, application sous la couche de prompt, isolement des identifiants hors du contexte IA — couvrent aussi les failles une et trois. L'implémentation MCP de Kiteworks applique des politiques par opération, valide les parcours d'accès et maintient les identifiants hors de portée du modèle. Le framework évolue avec la menace.

Trois questions pour situer votre niveau de sécurité

Faille 1 (Confiance dans l'entrée) : Validez-vous les données provenant de l'externe avant qu'elles ne soient traitées par l'IA ? Sinon, toutes les vulnérabilités décrites ici fonctionneraient dans votre environnement.

Faille 2 (Accès aux données) : Chaque requête de données IA est-elle évaluée par rapport à la politique, indépendamment — à chaque opération, et non au niveau de la session ? Sinon, une seule instruction injectée peut balayer tout votre périmètre de données.

Faille 3 (Confinement) : Savez-vous quelles API et quels canaux de sortie vos processus IA back-end peuvent invoquer ? Sinon, vous avez évalué l'accès aux données, mais pas le périmètre fonctionnel. Combiné à une entrée non fiable, c'est le scénario GrafanaGhost.

En savoir plus