AYI NEDJIMI Consultants

Attaques RAG : empoisonnement vector

16 mai 2026 • Mis à jour le 17 mai 2026 • 16 min de lecture • 3107 mots •

Maîtrisez les attaques contre les systèmes RAG : empoisonnement vectori
Défenses et hardening pour sécuriser vos bases vectorielles.

À RETENIR

A retenir - Attaques RAG et empoisonnement vectoriel

Les **attaques contre les systemes RAG** representent l'une des menaces les p
entreprise. Un attaquant capable d'injecter des documents malveillants dans
reponses du systeme, exfiltrer des donnees sensibles et contourner les guar
discret, persistant et difficile a detecter sans monitoring specialise. La defen
ingeres, le monitoring de la similarite cosine et l'isolation des sources de don

Les architectures **RAG (Retrieval-Augmented Generation)** sont devenues le stand
permettent d'ancrer les reponses du modele dans u sances me
cette puissance vient avec une surface d'attaque no t sous-estim

vectoriel exploitent la confiance implicite accordee aux documents recuperes par
d'injecter des chunks malveillants dans la base vectorielle peut influencer, manipu
systeme, transformant votre assistant IA metier en vecteur d'exfiltration ou de des
solutions RAG sur Pinecone, Qdrant, Weaviate et ChromaDB, la securisation de la
securite que tout RSSI doit integrer dans sa strategie de protection des systemes
documentees, les mecanismes sous-jacents, et les defenses concretes pour prote
vecteurs les plus frequemment exploites par les attaquants en 2026.

Architecture RAG et surface d'attaque -- vue d'ensemble

Un systeme RAG typique comprend quatre composants : le pipeline d'ingestion (c
vectorielle (stockage des embeddings), le retrieval engine (recherche de similarite
s'appuyant sur les chunks recuperes. Chacun de ces composants est potentiellem

La surface d'attaque se structure ainsi :

- **Pipeline d'ingestion** : injection de documents malveillants via les sources de do
  crawl, API tierces)
- **Base vectorielle** : acces direct a l'API de la vector DB pour insertion de vecteurs
  insuffisante
- **Retrieval** : manipulation de la requete d'embedding pour biaiser les resultats re
- **Contexte LLM** : injection d'instructions via les chunks recuperes (indirect prom

Pour aller plus loin sur les attaques d'injection indirecte dans les RAG, consultez n
poisoning. Les aspects de red teaming LLM complet incluant le testing RAG sont c
premise.

Réponse sous 24h

Devis gratuit →