AYI NEDJIMI Consultants

AI Model Supply Chain : Attaques sur Hugging Face et les

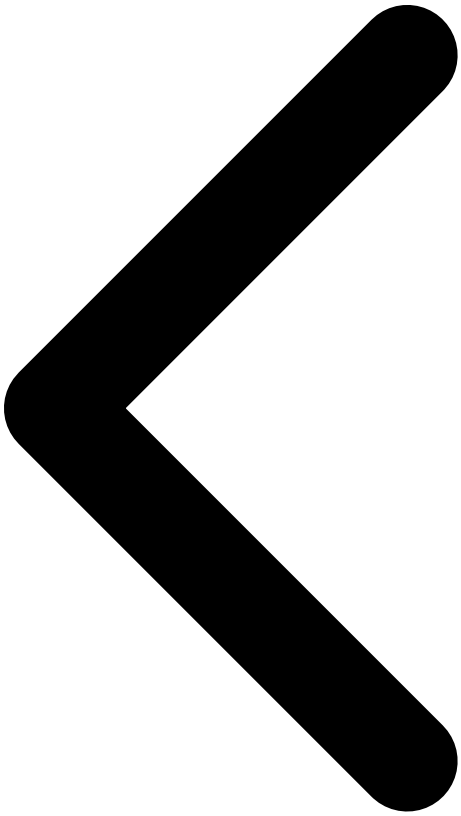
Catégorie : Intelligence Artificielle Lecture : 10 min Publié le : 28/02/2026 Auteur : Ayi NEDJIMI

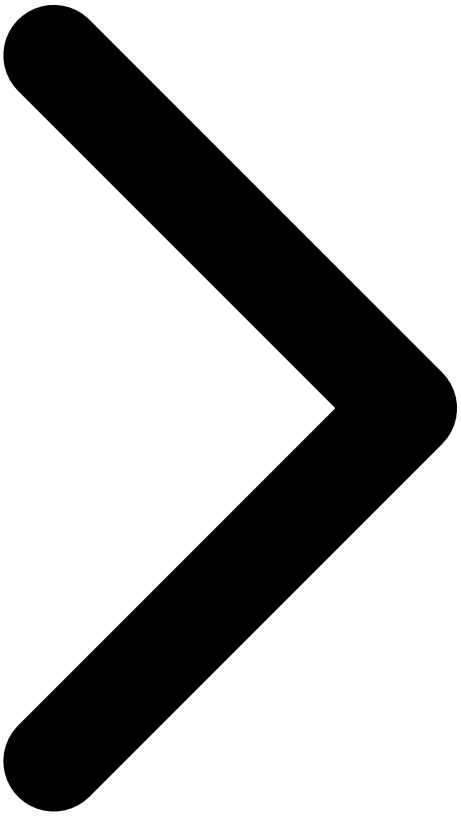
Risques des modèles pré-entraînés publics : pickle deserialization, backdoors dans les poids, typosquatting, scanning ModelScan et bonnes pratiques.

Le vecteur d'attaque le plus direct et le plus dangereux dans la supply chain des modèles IA est l'**exploitation de la désérialisation pickle**. Le format pickle de Python, utilisé historiquement par PyTorch pour sauvegarder les poids des modèles (fichiers .pt, .pth, .bin), permet l'exécution de code Python arbitraire lors du chargement. Un fichier pickle malveillant peut contenir des instructions qui, lors de l'appel à `torch.load()` ou `pickle.load()`, exécutent un reverse shell, téléchargent et installent un malware, exfiltrent des variables d'environnement contenant des credentials (clés API, tokens d'accès), ou modifient silencieusement d'autres fichiers du système. Risques des modèles pré-entraînés publics : pickle deserialization, backdoors dans les poids, typosquatting, scanning ModelScan et bonnes pratiques. Ce guide couvre les aspects essentiels de ia model supply chain huggingface : méthodologie structurée, outils recommandés et retours d'expérience opérationnels. Les professionnels y trouveront des recommandations directement applicables.

En réponse à ces risques, Hugging Face a développé le format **safetensors**, un format de sérialisation de tenseurs qui ne permet structurellement pas l'exécution de code. Safetensors utilise un format binaire simple : un header JSON contenant les métadonnées des tenseurs (noms, types, tailles) suivi des données brutes des tenseurs en mémoire contiguë. Aucun mécanisme de callback, d'objet personnalisé ou de code exécutable n'est possible dans ce format. Safetensors est également plus performant que pickle : le chargement est jusqu'à 10x plus rapide grâce au memory-mapping, et la validation d'intégrité est intégrée. Depuis 2024, Hugging Face affiche un avertissement pour tout modèle publié au format pickle et encourage la migration vers safetensors. Cependant, des centaines de milliers de modèles existants restent au format pickle, et de nombreux workflows continuent d'utiliser `torch.load()` par habitude. Pour approfondir, consultez Apprentissage Fédéré et Privacy-Preserving ML en Cybersécurité.

Au-delà de pickle, d'autres formats posent des risques similaires. Les fichiers **ONNX** peuvent contenir des opérateurs personnalisés (custom ops) qui exécutent du code natif. Les **fichiers de configuration** (config.json, tokenizer_config.json) peuvent référencer du code Python personnalisé via le mécanisme `trust_remote_code=True` de la bibliothèque transformers — un flag qui charge et exécute des fichiers Python arbitraires depuis le dépôt du modèle. Les **Jupyter notebooks** inclus dans les dépôts de modèles peuvent contenir du code malveillant exécuté lors de l'ouverture. Et les **scripts d'entraînement** accompagnant les modèles peuvent contenir des payloads dissimulés dans des commentaires encodés ou des variables non évidentes.





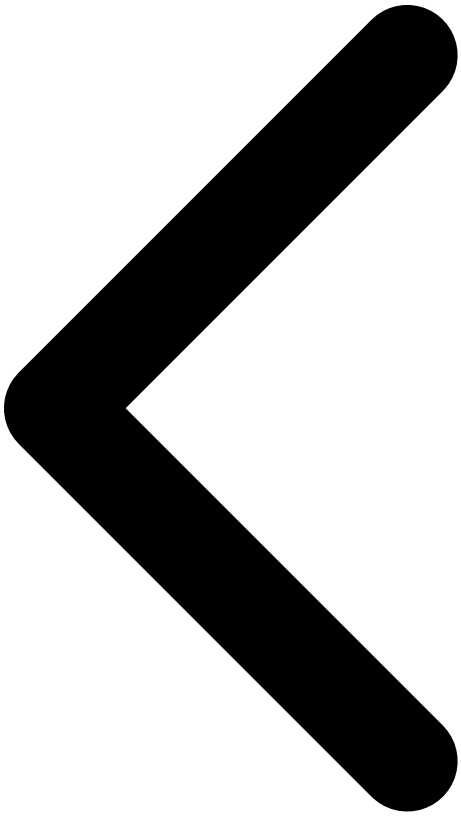
Notre avis d'expert
Chez Ayi NEDJIMI Consultants, nous constatons que la majorité des organisations sous-estiment les risques liés aux modèles de langage déployés en production. La sécurité des LLM ne se limite pas au prompt engineering : elle exige une approche systémique couvrant les embeddings, les pipelines de données et les mécanismes de contrôle d'accès aux API.

3 Backdoors dans les poids des modèles

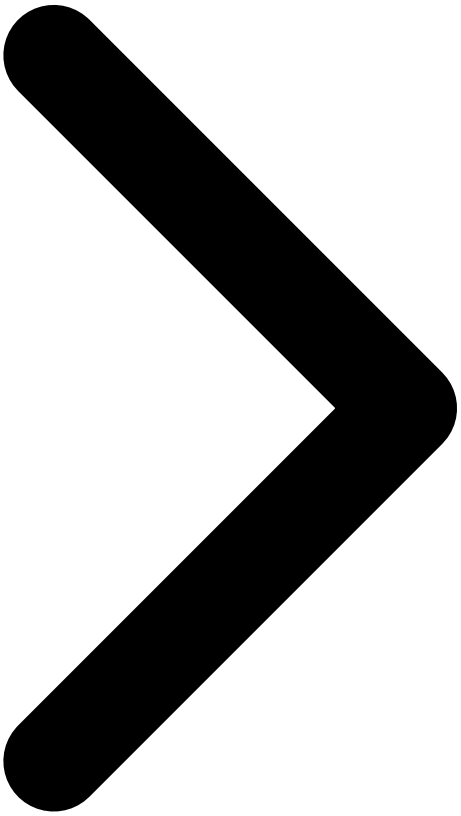
Les **backdoors dans les poids** représentent une menace plus subtile et plus difficile à détecter que l'exécution de code via pickle. Contrairement aux attaques de désérialisation qui agissent au niveau du système, les backdoors de poids opèrent au niveau du comportement du modèle lui-même. Un modèle backdooré se comporte normalement dans la grande majorité des cas, mais produit un comportement malveillant spécifique quand un **trigger** prédéfini est présent dans l'entrée.

Les techniques d'insertion de backdoors incluent le **data poisoning** (injection de données d'entraînement contenant le trigger associé à la sortie malveillante souhaitée), le **weight manipulation** (modification directe des poids après entraînement pour insérer le comportement trigger), et le **fine-tuning malveillant** (fine-tuning d'un modèle sain sur un dataset contenant des exemples backdoorés). Les triggers peuvent être textuels (un mot ou une phrase spécifique), visuels (un pattern de pixels dans une image), ou structurels (un pattern syntaxique dans du code). Les backdoors les plus avancées utilisent des triggers **distribués** : aucun élément individuel n'est le trigger, c'est la combinaison de plusieurs caractéristiques subtiles qui active le comportement malveillant.

La détection des backdoors de poids est un problème de recherche actif. Les approches incluent le **Neural Cleanse** (identification du trigger minimal qui provoque une classification uniforme), le **Activation Clustering** (analyse des activations internes pour détecter les neurones associés au backdoor), le **STRIP** (perturbation de l'input et observation de la stabilité de la prédiction — les inputs backdoorés sont anormalement stables), et le **Meta Neural Analysis** (entraînement d'un meta-classifieur qui distingue les modèles backdoorés des modèles sains à partir de leurs caractéristiques de poids). Aucune de ces techniques n'offre de garantie complète, et les backdoors élaborées restent extrêmement difficiles à détecter en pratique.



Vecteurs d'attaque Backdoors Typosquatting



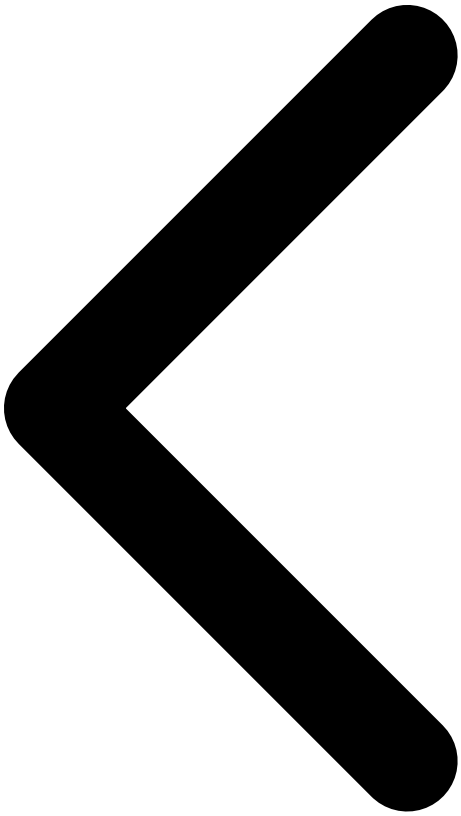
Comment garantir que vos modèles de machine learning ne deviennent pas des vecteurs d'attaque ?

4 Typosquatting de modèles

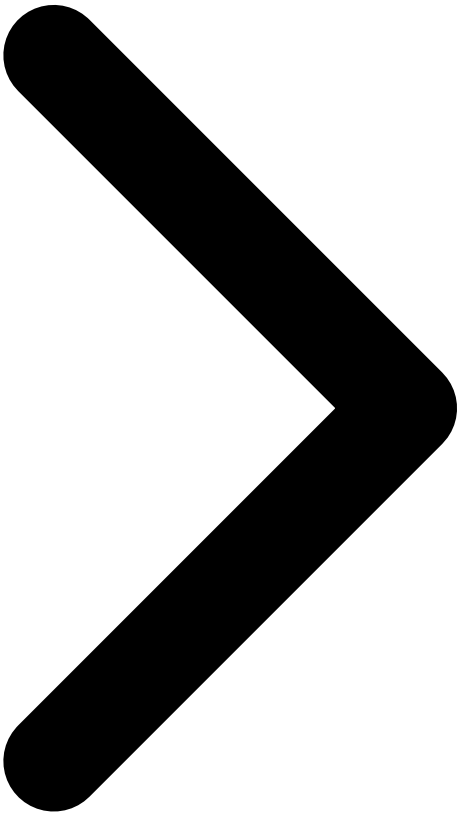
Le **typosquatting de modèles** transpose une technique d'attaque bien connue du monde des packages logiciels (npm, PyPI) à l'écosystème des modèles IA. L'attaquant publie un modèle dont le nom est proche d'un modèle populaire — par exemple `meta-llama/Llama-2-7b-chat-hf` devient `meta-Ilama/Llama-2-7b-chat-hf` (avec un I majuscule à la place du l minuscule) ou `meta-llama/LLama-2-7b-chat-hf`. Le modèle typosquatté peut contenir un payload malveillant (pickle exploit), des poids backdoorés, ou simplement un modèle de mauvaise qualité présenté comme le modèle légitime.

En 2025, des chercheurs de JFrog ont identifié plus de 100 modèles malveillants sur Hugging Face utilisant des techniques de typosquatting, de dependency confusion et de namespacesquatting. Certains de ces modèles avaient accumulé des milliers de téléchargements avant d'être détectés et retirés. Les attaquants exploitent également le **star**

manipulation (faux comptes donnant des étoiles pour augmenter la visibilité) et le **readme manipulation** (descriptions copiées des modèles légitimes pour augmenter la crédibilité). La protection contre le typosquatting repose sur la vérification systématique de l'organisation et de l'auteur du modèle, l'utilisation de hashes de vérification, et la mise en place de registres de modèles internes avec des processus de validation. Pour approfondir, consultez Comment Choisir sa Base.



Backdoors Typosquatting Supply chain MLOps

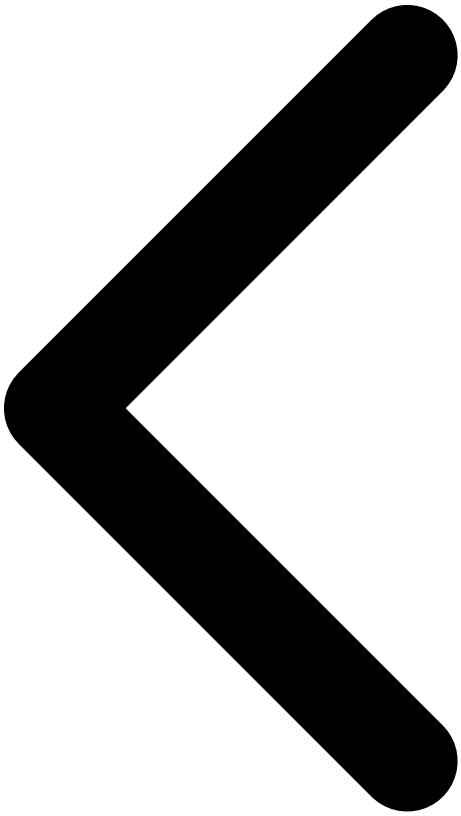


Cas concret
En février 2024, une entreprise de Hong Kong a perdu 25 millions de dollars après qu'un employé a été trompé par un deepfake vidéo lors d'une visioconférence. Les attaquants avaient recréé l'apparence et la voix du directeur financier à l'aide de modèles d'IA générative, démontrant les risques concrets de cette technologie en contexte corporate.

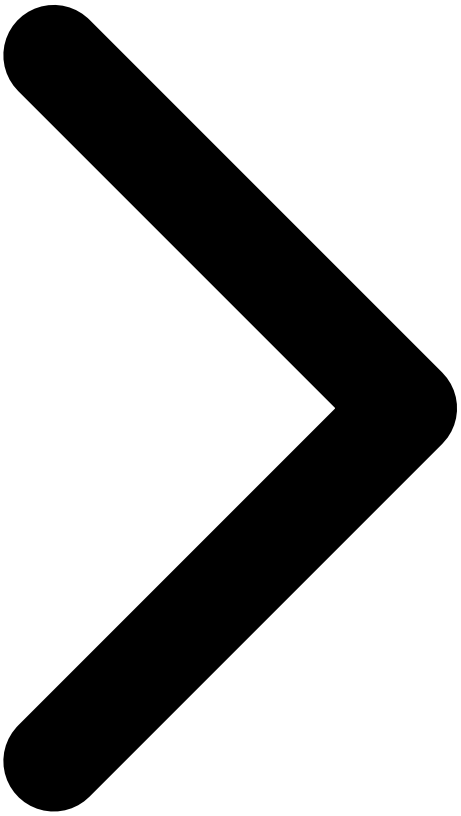
5 Supply chain MLOps

La supply chain des modèles IA s'étend bien au-delà des fichiers de poids. L'ensemble du pipeline **MLOps** — de la collecte des données d'entraînement au déploiement en production — présente des points de vulnérabilité. Les **datasets publics** (Common Crawl, The Pile, RedPajama) peuvent être empoisonnés par injection de données malveillantes dans les sources web indexées. Les **bibliothèques ML** (transformers, pytorch, tensorflow) sont des dépendances critiques dont la compromission affecterait des millions de déploiements. Les **pipelines CI/CD** pour le ML (MLflow, Kubeflow, Weights & Biases) manipulent des artefacts de modèles et des credentials d'accès aux registres.

Le concept de **ML Bill of Materials (ML-BOM)** émerge comme réponse structurée à ces défis. Par analogie avec le Software Bill of Materials (SBOM) pour les logiciels, un ML-BOM documente l'ensemble des composants d'un système ML : modèle de base utilisé, datasets d'entraînement et de fine-tuning, bibliothèques et versions, hyperparamètres d'entraînement, métriques d'évaluation, et provenance de chaque composant. Les formats émergents incluent le **Model Card** (Hugging Face), la **AI Factsheet** (IBM), et le **Supply-chain Levels for Software Artifacts (SLSA)** adapté au ML. L'AI Act européen impose des exigences de traçabilité qui rendront les ML-BOM obligatoires pour les systèmes IA à haut risque déployés dans l'UE.



Typosquatting Supply chain MLOps Scanning



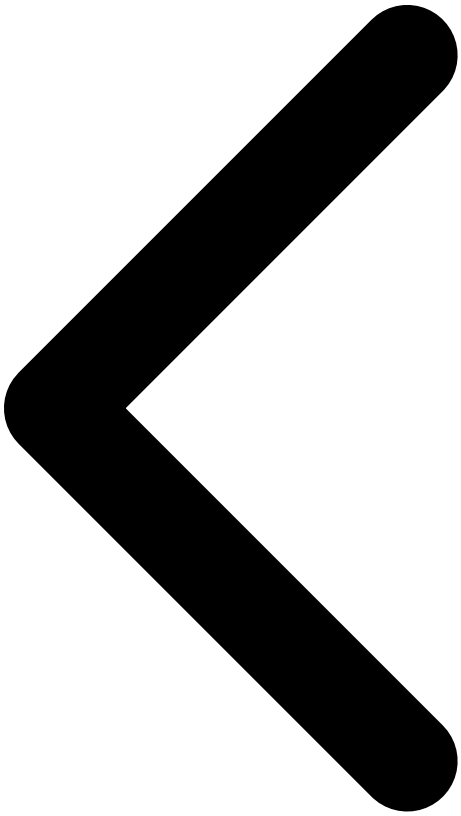
6 Scanning et vérification (ModelScan)

Plusieurs outils ont émergé pour scanner les modèles IA à la recherche de menaces. **ModelScan**, développé par Protect AI, est l'outil de référence pour la détection de code malveillant dans les fichiers de modèles. Il analyse les fichiers pickle, ONNX, Keras et autres formats pour détecter les appels de fonctions dangereuses (os.system, subprocess, eval, exec), les reverse shells, les téléchargeurs de malware, et les patterns de code obfusqué. ModelScan fonctionne par analyse statique du bytecode pickle sans exécution, éliminant le risque d'infection lors du scan.

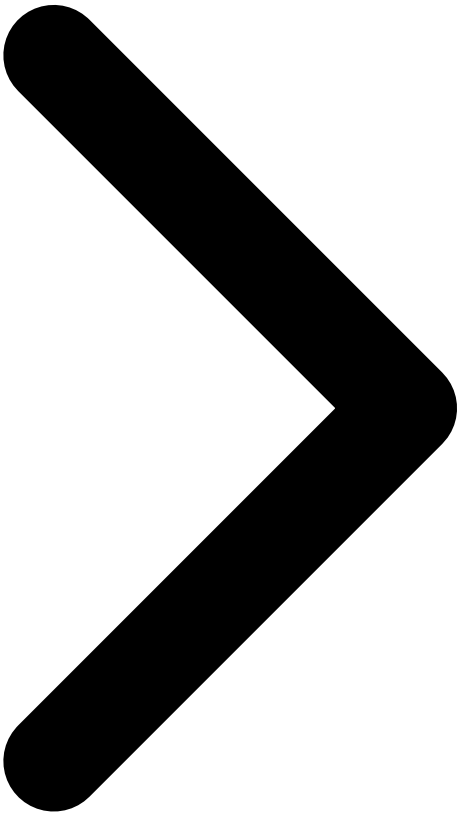
Hugging Face Security Scanner intègre désormais un scanning automatique de tous les modèles publiés sur la plateforme. Les modèles détectés comme malveillants sont signalés par un badge d'avertissement et peuvent être retirés. Le scanner analyse les fichiers pickle pour le code malveillant, vérifie la cohérence entre les fichiers de poids et la configuration déclarée, et

détecte les patterns de typosquatting. **Fickling**, développé par Trail of Bits, est un outil de décompilation pickle qui permet d'inspecter le code contenu dans un fichier pickle avant de le charger, offrant une visibilité sur les opérations qui seront exécutées.

L'intégration de ces outils dans le pipeline MLOps est essentielle. La configuration recommandée inclut un **scan ModelScan automatique** dans le pipeline CI/CD à chaque import de modèle externe, une politique de **safetensors-only** rejetant tout modèle au format pickle, une **vérification de signatures** pour les modèles provenant d'organisations de confiance (Hugging Face supporte les signatures GPG pour les organisations vérifiées), et un **registre de modèles interne** (via MLflow Model Registry, DVC, ou un registry OCI) servant de sas de validation entre les modèles publics et l'infrastructure de production.



Supply chain Scanning Bonnes pratiques



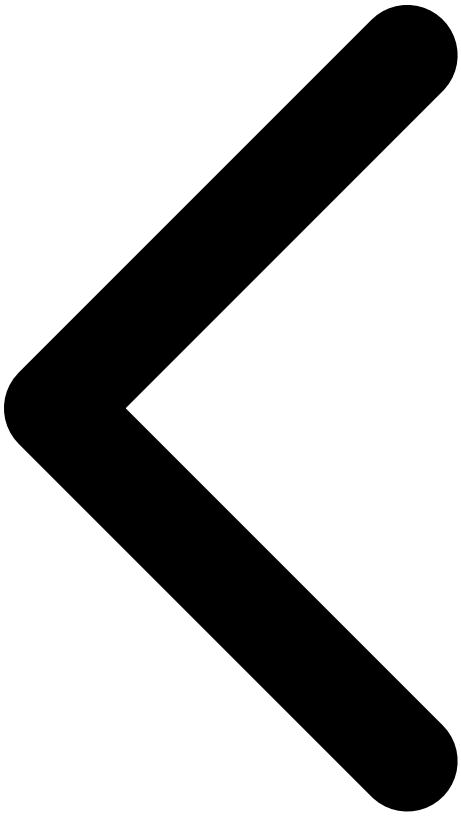
7 Bonnes pratiques de provenance

La sécurisation de la supply chain des modèles IA exige une approche systématique couvrant l'ensemble du cycle de vie. La **provenance des modèles** doit être documentée et vérifiable à chaque étape. Pour les modèles externes, privilégiez les sources officielles : les organisations vérifiées sur Hugging Face (badge bleu), les releases officielles sur GitHub des développeurs du modèle, et les registres d'entreprise (AWS Marketplace, Azure AI Gallery, Google Cloud AI Platform). Vérifiez systématiquement le hash SHA-256 des fichiers téléchargés contre les hashes publiés par le fournisseur. Pour approfondir, consultez OWASP Top 10 pour les LLM : Guide Remédiation 2026.

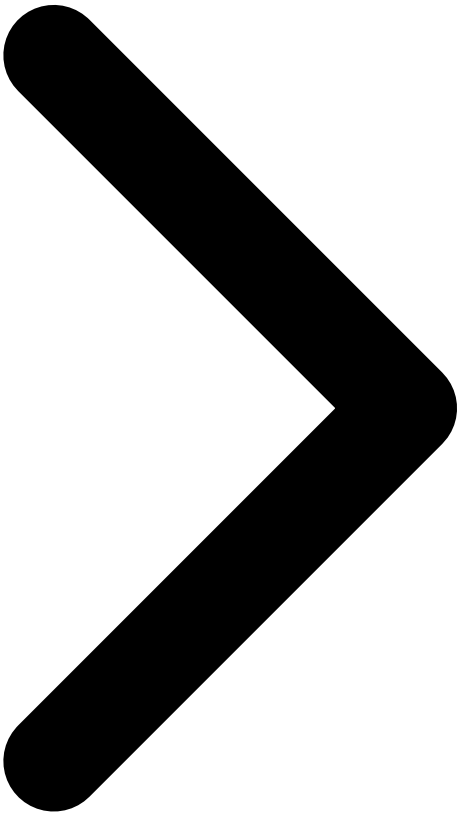
Pour les modèles utilisés en production, implémentez un **processus de validation en plusieurs étapes** : scan automatique ModelScan et Fickling à l'import, évaluation sur un benchmark de sécurité (détection de backdoors via Neural Cleanse ou STRIP), revue manuelle des fichiers de configuration et du code personnalisé, test de non-régression sur les métriques métier, et approbation formelle avant déploiement. Maintenez un **inventaire centralisé** de tous les

modèles déployés avec leur provenance, version, hash, date de déploiement et propriétaire métier. En cas de découverte d'une vulnérabilité dans un modèle de base, cet inventaire permet d'identifier rapidement tous les déploiements affectés.

- ▹**Format safetensors :** rejeter systématiquement les modèles au format pickle, imposer safetensors comme standard
- ▹**Scanning automatique :** intégrer ModelScan dans le pipeline CI/CD pour tout modèle importé
- ▹**trust_remote_code=False :** ne jamais activer l'exécution de code distant sans revue de sécurité complète
- ▹**Registre interne :** utiliser un registry de modèles interne comme sas de validation obligatoire
- ▹**ML-BOM :** documenter la provenance complète de chaque modèle (base, datasets, fine-tuning, versions)



Scanning Bonnes pratiques Conclusion



8 Conclusion et recommandations

La sécurité de la **supply chain des modèles IA** est un enjeu critique qui ne fera que croître avec la prolifération des modèles pré-entraînés et la complexification des pipelines MLOps. Les risques sont concrets et documentés : exécution de code arbitraire via pickle, backdoors indétectables dans les poids, typosquatting de modèles populaires, et empoisonnement des datasets d'entraînement. Les organisations qui déploient des modèles IA sans processus de validation s'exposent à des compromissions potentiellement critiques.

L'écosystème de sécurité s'organise rapidement : le format safetensors élimine le risque pickle, ModelScan et Fickling permettent le scanning automatisé, Hugging Face renforce sa détection des modèles malveillants, et les standards de provenance (ML-BOM, SLSA for ML) se structurent. Les organisations doivent adopter ces outils et pratiques sans attendre, en les intégrant dans leurs processus de gouvernance IA existants et en formant leurs équipes ML aux risques spécifiques de la supply chain.

Action immédiate : Auditez dès maintenant votre inventaire de modèles en production. Identifiez ceux au format pickle et planifiez leur migration vers safetensors. Intégrez ModelScan dans vos pipelines CI/CD. Et établissez une politique claire interdisant le `trust_remote_code=True` sans approbation de sécurité explicite.

Besoin d'un accompagnement expert ?

Nos consultants en cybersécurité et IA vous accompagnent dans la sécurisation de votre supply chain MLOps. Devis personnalisé sous 24h.

Références et ressources externes

- OWASP LLM Top 10 — Les 10 risques majeurs pour les applications LLM
- MITRE ATT&CK T1195 — Supply Chain Compromise
- NIST AI RMF — AI Risk Management Framework du NIST
- arXiv — Archive ouverte de publications scientifiques en IA
- HuggingFace Docs — Documentation de référence pour les modèles de ML

Pour approfondir ce sujet, consultez notre outil open-source ml-model-security-audit qui facilite l'évaluation de la sécurité des modèles ML.

Sources et références : ArXiv IA · Hugging Face Papers

> Points clés à retenir
>
> - 3 Backdoors dans les poids des modèles
> - 4 Typosquatting de modèles
> - 5 Supply chain MLOps
> - 6 Scanning et vérification (ModelScan)
> - 7 Bonnes pratiques de provenance
> - 8 Conclusion et recommandations

FAQ

Qu'est-ce que AI Model Supply Chain ?

Le concept de AI Model Supply Chain est détaillé dans les premières sections de cet article, qui couvrent les fondamentaux, les enjeux et le contexte opérationnel. Pour un accompagnement sur ce sujet, contactez nos experts.

Pourquoi AI Model Supply Chain est-il important en cybersécurité ?

La compréhension de AI Model Supply Chain permet aux équipes de sécurité d'améliorer leur posture défensive. Les sections « 3 Backdoors dans les poids des modèles » et « 4 Typosquatting de modèles » détaillent les raisons de cette importance. Pour un accompagnement sur ce sujet, contactez nos experts.

Comment mettre en œuvre les recommandations de cet article ?

Les recommandations pratiques sont détaillées tout au long de l'article, avec des commandes, des outils et des méthodologies éprouvées. La section « Conclusion » fournit une synthèse actionnable. Pour un accompagnement sur ce sujet, contactez nos experts.

Conclusion

Cet article a couvert les aspects essentiels de Table des Matières, 1 Introduction : la supply chain des modèles IA, 2 Vecteurs d'attaque (pickle deserialization, safetensors). La mise en pratique de ces recommandations permet de renforcer significativement la posture de securite de votre organisation.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr