Agents LLM pour le Pentest : 10 Outils Open Source 2026

En 2026, une dizaine d'outils open source permettent à des agents LLM (GPT-4o, Claude, Llama) d'automatiser des phases entières de pentest : reconnaissance, énumération, exploitation web et escalade de privilèges. Ces agents ne remplacent pas le pentester senior, mais ils accélèrent massivement les phases répétitives et peuvent trouver des vulnérabilités que les outils classiques ratent. Ce guide compare les 10 outils les plus actifs, leurs architectures, leurs performances sur XBOW et HackTheBox, et les conditions légales d'utilisation en France.

Le pentest automatisé par intelligence artificielle n'est plus une promesse de laboratoire. En 2026, des équipes de recherche universitaires, des bug bounty hunters et des red teamers professionnels utilisent quotidiennement des agents LLM pour conduire des tests d'intrusion sur des cibles autorisées. Le mouvement a débuté en 2023 avec les premiers travaux académiques sur GPT-4 et la capacité des grands modèles de langage à comprendre des sorties de scanners réseau, à interpréter des CVE et à générer des payloads adaptés au contexte. Depuis, l'écosystème a explosé : PentestGPT a publié un papier à USENIX Security, XBOW a créé le premier benchmark standardisé pour agents de pentest autonomes, et une nouvelle génération d'outils comme PentesterFlow, HexStrike AI, pwnkit et open·kritt ont atteint une maturité suffisante pour des usages professionnels. Ce guide s'adresse aux pentesters, RSSI, consultants en cybersécurité et chercheurs qui souhaitent comprendre l'état de l'art, évaluer les outils disponibles et intégrer les agents LLM dans leurs workflows. Nous couvrons en détail chaque outil — architecture, capacités, limites, modèles supportés — ainsi que les benchmarks de performance, le cadre légal français, les risques de confidentialité liés à l'envoi de données de scan à des API cloud, et les facteurs qui déterminent le choix du bon agent selon le contexte de mission. Avertissement préalable : tous les outils décrits dans cet article doivent être utilisés exclusivement sur des systèmes pour lesquels une autorisation écrite explicite a été obtenue. L'utilisation sur des systèmes tiers sans autorisation constitue une infraction au Code pénal français (article 323-1) passible de trois ans d'emprisonnement et 45 000 € d'amende.

À RETENIR

⚡ À retenir — Agents LLM pour le Pentest

Les agents LLM accélèrent les phases répétitives du pentest (recon, énumération, fuzzing) mais ne remplacent pas le jugement d'un expert humain. Les meilleurs résultats s'obtiennent en mode human-in-the-loop, pas en autonomie totale. Un pentest GPT-4o peut coûter 50 à 500 € en tokens — les modèles locaux via Ollama sont préférables pour les cibles sensibles. PentestGPT (USENIX 2024) et CAI dominent en maturité ; PentesterFlow et HexStrike AI sont les plus adaptés aux workflows professionnels. L'autorisation écrite préalable est une obligation légale absolue en France — les outils IA ne changent pas ce cadre.

Pourquoi les agents LLM révolutionnent le pentest en 2026

Le pentest traditionnel repose sur un cycle bien établi : reconnaissance passive, énumération active, identification de vulnérabilités, exploitation, post-exploitation, reporting. Chacune de ces phases mobilise des outils spécialisés — nmap, gobuster, sqlmap, Metasploit, BloodHound — et requiert qu'un expert humain interprète les sorties, choisisse la prochaine action et adapte son approche au contexte de la cible. C'est précisément ce processus d'interprétation et de décision contextuelle que les grands modèles de langage ont commencé à automatiser avec une efficacité surprenante.

La rupture est intervenue en 2023 quand des chercheurs ont démontré que GPT-4, sans fine-tuning spécifique, pouvait résoudre des challenges HackTheBox de niveau débutant à intermédiaire en chaînant des appels d'outils. Le modèle comprenait les sorties de nmap, inférait les services exposés, recherchait des CVE pertinentes et proposait des vecteurs d'exploitation cohérents. Ce n'était pas de la magie : c'était la conséquence directe d'un entraînement massif sur des données de sécurité — writeups CTF, rapports de pentest, documentation des outils, papiers académiques sur les vulnérabilités.

En 2026, les agents LLM offensifs présentent plusieurs avantages distincts par rapport aux outils traditionnels. Premièrement, ils s'adaptent au contexte : là où sqlmap lance une batterie de payloads standards, un agent LLM peut analyser la logique applicative, identifier des paramètres non évidents et générer des payloads personnalisés. Deuxièmement, ils gèrent la complexité multi-étapes : une chaîne SSRF → accès métadonnées cloud → exfiltration de clés IAM → escalade de privilèges AWS peut être planifiée et exécutée de bout en bout sans intervention humaine à chaque étape. Troisièmement, ils documentent automatiquement leur travail : les étapes d'exploitation, les payloads utilisés et les preuves collectées sont logués de façon structurée, ce qui accélère considérablement la rédaction des rapports.

Cela dit, les limites sont réelles et bien documentées. Les agents LLM génèrent des faux positifs en quantité significative — ils "voient" des vulnérabilités qui n'existent pas ou confondent des comportements normaux avec des indicateurs d'exploitabilité. Ils peuvent halluciner des CVE, proposer des payloads syntaxiquement corrects mais sémantiquement inefficaces, ou abandonner une piste prometteuse parce que la sortie d'un outil dépasse leur fenêtre de contexte. C'est pourquoi le mode human-in-the-loop — où l'agent propose et l'humain valide avant exécution — reste la référence pour des missions professionnelles où la qualité du rapport importe autant que la découverte des vulnérabilités.

Le marché en 2026 se segmente clairement : des outils académiques orientés recherche et démonstration de faisabilité (HackingBuddyGPT, les premières versions de PentestGPT), des plateformes professionnelles avec reporting structuré (PentesterFlow, CyberStrike AI), et des frameworks d'intégration qui permettent à n'importe quel LLM d'accéder à un arsenal d'outils existants (HexStrike AI via MCP). Comprendre cette segmentation est la première étape pour choisir le bon outil selon son contexte de mission.

Architecture des agents offensifs : ReAct, tool-calling et planification autonome

Avant de comparer les outils, il est utile de comprendre les architectures sous-jacentes, car elles déterminent directement les capacités et les limites de chaque agent.

Le paradigme ReAct (Reasoning + Acting, Yao et al. 2022) est le plus répandu. L'agent alterne des étapes de raisonnement (chain-of-thought interne) et d'action (appel d'outil). À chaque itération, il observe le résultat de l'action précédente, met à jour son raisonnement et décide de la prochaine action. C'est le modèle utilisé par PentestGPT et HackingBuddyGPT dans leurs versions initiales.

Le tool-calling structuré (function calling d'OpenAI, tool_use d'Anthropic) permet au LLM d'émettre des appels d'outils formatés en JSON, que le framework exécute et dont il retourne les résultats au modèle. C'est plus fiable que le ReAct pur car les appels d'outils sont validés syntaxiquement avant exécution. HexStrike AI via MCP utilise cette approche.

La planification multi-étapes avec arbre de tâches est l'architecture la plus avancée : l'agent décompose l'objectif de haut niveau (« trouver une RCE sur cette application ») en sous-tâches ordonnées (recon → enum → test injection → exploitation), puis exécute chaque branche avec un sous-agent spécialisé. CAI et CyberStrike AI implémentent cette architecture avec des agents spécialisés par phase.

Le pipeline parallèle de PENTDEM pousse plus loin : plusieurs outils s'exécutent en parallèle dans des conteneurs Docker isolés, et l'agent agrège les résultats pour identifier les pistes les plus prometteuses. Cette approche maximise la couverture au prix d'une complexité d'orchestration et d'un coût en tokens plus élevé.

Un facteur critique souvent négligé est la gestion de la fenêtre de contexte. Les sorties de scan réseau (nmap sur 65535 ports), de fuzzing web (gobuster sur une wordlist de 100 000 entrées) ou d'analyse de code source peuvent dépasser la fenêtre de contexte de n'importe quel modèle actuel. Les agents matures implémentent des mécanismes de résumé et de filtrage — ne transmettre que les lignes pertinentes au LLM — ce qui réduit les coûts et améliore la qualité du raisonnement.

PentestGPT : le pionnier académique (USENIX Security 2024)

PentestGPT est l'outil qui a mis les agents LLM de pentest sur la carte académique. Développé par Gelei Deng et ses collègues de l'Université Nationale de Singapour, il a été présenté au

USENIX Security Symposium 2024 et a établi une référence empirique sur les capacités des LLMs en pentest autonome. Le code source est disponible sur GitHub (GreyDGL/PentestGPT).

L'architecture de PentestGPT repose sur trois modules LLM spécialisés : un *reasoning module* qui maintient le contexte global du pentest et décide des prochaines étapes, un *generation module* qui produit les commandes spécifiques à exécuter, et un *parsing module* qui interprète les sorties d'outils et les intègre dans le contexte. Cette séparation des responsabilités améliore la cohérence des décisions et limite les effets de dilution de contexte.

Les résultats publiés dans le papier original sont significatifs : PentestGPT résout 64 % des challenges HackTheBox de niveau « easy » sans intervention humaine, contre 41 % pour l'utilisation directe de GPT-4 sans orchestration. Sur les challenges « medium », le taux tombe à 35 %, ce qui illustre bien la limite actuelle des agents face à des scénarios nécessitant des raisonnements multi-étapes complexes ou une créativité offensive que les LLMs n'ont pas encore internalisée.

La version **Agentic v1.0**, publiée mi-2025, apporte une autonomie significativement accrue. L'agent peut désormais enchaîner des dizaines d'actions sans intervention humaine, avec une gestion améliorée des dead-ends (l'agent reconnaît quand une piste est épuisée et bascule sur une autre) et un système de mémoire externe qui persiste les informations découvertes entre les sessions. Cette version est particulièrement efficace sur les machines HTB de type « realistic » qui simulent des environnements d'entreprise avec Active Directory, services web et accès API.

Pour les professionnels, PentestGPT est surtout utile comme outil de formation et de recherche. Son code bien documenté permet de comprendre comment architecturer un agent offensif, et ses logs détaillés offrent une visibilité unique sur le raisonnement du LLM face à des situations de pentest réelles. En mission professionnelle, on lui préférera des outils avec reporting structuré intégré, mais PentestGPT reste la référence académique incontournable pour évaluer les capacités brutes des LLMs en contexte offensif.

PentesterFlow : l'agent human-in-the-loop pour le bug bounty

PentesterFlow adopte une philosophie radicalement différente de PentestGPT : plutôt que de maximiser l'autonomie, il optimise la collaboration entre l'agent et le pentester humain. C'est un outil CLI qui planifie les étapes du test, exécute les outils offensifs, mais demande systématiquement une validation humaine avant toute action potentiellement destructive ou irréversible. Cette approche human-in-the-loop n'est pas une limitation — c'est une feature délibérée qui le rend adapté aux contextes professionnels où la maîtrise du processus est aussi importante que les résultats.

L'outil est spécialisé dans le pentest web et le bug bounty, avec des *skills* dédiés aux vulnérabilités les plus courantes dans ce contexte : IDOR (Insecure Direct Object References), SSRF (Server-Side Request Forgery), injection JWT, abus GraphQL et manipulation de logique métier. Pour chaque vecteur, PentesterFlow dispose d'une bibliothèque de techniques testées et d'une logique de décision contextuelle — par exemple, détecter automatiquement si une API GraphQL expose un endpoint d'introspection et adapter les queries en conséquence.

Le reporting est l'un des points forts différenciants : les rapports sont générés en Markdown avec des sections PoC (Proof of Concept) reproduisibles, des étapes de remédiation et des références CVE/CWE. Le format est directement utilisable dans des plateformes de bug bounty comme HackerOne ou Bugcrowd, ce qui représente un gain de temps considérable pour les hunters professionnels.

Sur le plan des modèles supportés, PentesterFlow est l'un des rares outils à proposer un support natif d'**Ollama** pour les modèles locaux (Llama 3, Mistral, Qwen), ce qui le rend particulièrement pertinent pour les missions sur des cibles sensibles où envoyer les sorties de scan à une API cloud serait problématique. Publié en open source en juillet 2026, il est déjà bien adopté par la communauté bug bounty francophone.

HexStrike AI : piloter 150 outils de sécurité via MCP

HexStrike AI prend une approche architecturale distincte de tous les autres outils de cette liste : c'est un **serveur MCP** (Model Context Protocol), le standard open source développé par Anthropic pour permettre aux LLMs d'accéder à des outils externes via une interface unifiée. Au lieu de créer un nouvel agent LLM, HexStrike expose plus de 150 outils de cybersécurité — nmap, nikto, sqlmap, ffuf, amass, nuclei, subfinder, gobuster, wpscan, et bien d'autres — comme des tools MCP que n'importe quel client compatible peut utiliser.

La conséquence pratique est remarquable : vous pouvez utiliser HexStrike avec Claude (via l'interface web ou l'API), avec GPT-4o, avec Gemini 1.5 Pro, ou avec tout autre LLM compatible MCP. L'agent IA décide quels outils appeler selon le contexte, les enchaîne intelligemment et agrège les résultats. La flexibilité est maximale, et il n'y a pas de code agent custom à maintenir côté utilisateur.

Cette architecture présente cependant une limite importante pour les missions professionnelles : la qualité du raisonnement dépend entièrement du LLM choisi. Claude et GPT-4o offrent les meilleurs résultats, mais les modèles locaux via Ollama peuvent produire des hallucinations plus fréquentes sur les décisions d'orchestration d'outils. HexStrike est particulièrement adapté à la recherche de vulnérabilités, au bug bounty et à la phase de reconnaissance initiale, où la couverture des outils est plus importante que la précision de l'exploitation.

L'outil est particulièrement pertinent pour les équipes qui travaillent déjà avec des LLMs via l'API Anthropic ou OpenAI : l'intégration d'HexStrike ajoute immédiatement une dimension offensive à leurs workflows existants sans nécessiter de refonte architecturale. Pour un audit de cybersécurité PME, HexStrike peut automatiser toute la phase de reconnaissance externe en quelques heures là où un humain y passerait une journée.

pwnkit : l'agent autonome aux scores XBOW

pwnkit se distingue par ses performances sur le **benchmark XBOW**, le premier référentiel standardisé pour évaluer objectivement les agents de pentest autonomes, créé par la société

XBOW Security en 2025. Ce benchmark évalue les agents sur des environnements d'exploitation réalistes — applications web vulnérables, APIs REST avec des failles logiques, instances cloud mal configurées — en mesurant le taux de réussite, le temps d'exploitation et la qualité des artefacts générés.

L'architecture de pwnkit est centrée sur l'exploitation des applications web et des APIs. L'agent maintient un état persistant de la cible — surface d'attaque découverte, vecteurs testés, résultats — et utilise cet état pour guider ses décisions d'exploration. La particularité « artefact-backed » signifie que chaque étape d'exploitation est accompagnée d'une preuve vérifiable : captures des requêtes/réponses HTTP, screenshots, fichiers exfiltrés. Cette traçabilité est critique pour des rapports de pentest crédibles.

pwnkit est l'un des agents les plus autonomes de cette liste, avec un niveau d'intervention humaine minimale requis. Pour des environnements bien définis et bornés — une application web en scope de bug bounty, un environnement de test isolé — cette autonomie est un avantage. Pour des missions en environnement de production, la prudence s'impose : un agent autonome sans supervision peut provoquer des effets de bord non intentionnels (déni de service par fuzzing excessif, génération d'alertes SOC, corruption de données de test).

CyberStrike AI : l'orchestrateur AGPL à 100+ outils

CyberStrike AI (également nommé CyberStrikeAI sur certains dépôts) est la plateforme la plus complète en termes d'architecture d'orchestration. Elle s'appuie sur un modèle de rôles spécialisés : un agent recon, un agent énumération, un agent exploitation, un agent post-exploitation et un agent reporting travaillent en coordination sous la supervision d'un orchestrateur central. Cette spécialisation améliore la qualité des décisions dans chaque phase, au prix d'une complexité de configuration plus élevée.

Avec plus de 100 outils intégrés via un système de plugins, CyberStrike AI couvre l'ensemble du cycle de pentest — de la reconnaissance OSINT passive à l'exploitation de vulnérabilités connues, en passant par le fuzzing web, le scanning de configurations cloud et la détection de

secrets exposés. Le système de *skills* permet d'ajouter des modules personnalisés, ce qui le rend extensible selon les besoins spécifiques d'une organisation.

La **licence AGPL** est un point important à connaître : toute organisation qui déploie CyberStrike AI dans un service accessible à des tiers (SaaS pentest automatisé, par exemple) doit publier son code source modifié. Pour un usage interne dans une équipe de sécurité, l'AGPL ne pose pas de contrainte pratique. Le reporting automatisé est l'un des points forts : CyberStrike génère des rapports exécutifs et techniques structurés, avec des niveaux de sévérité CVSS, des recommandations de remédiation et des preuves d'exploitation — un gain de temps considérable en fin de mission.

HackingBuddyGPT : le framework minimaliste pour chercheurs

HackingBuddyGPT adopte une philosophie diamétralement opposée à CyberStrike AI : la minimalisme assumé. L'objectif des auteurs n'est pas de créer le meilleur outil de pentest, mais de fournir un framework de recherche aussi simple que possible pour étudier comment les LLMs raisonnent dans des contextes d'attaque. Un agent basique peut être implémenté en **moins de 50 lignes de code Python**, ce qui facilite la compréhension et la modification.

Les cas d'usage principaux sont l'**escalade de privilèges sous Linux** et le **pivoting Active Directory**. L'agent reçoit un accès shell avec des privilèges limités et doit, en autonomie, identifier un vecteur d'escalade (SUID binaires, cron mal configuré, capabilities Linux, secrets en clair) et l'exploiter. Sur les machines HTB de niveau débutant à intermédiaire, les résultats sont corrects. Sur des environnements plus complexes, les limites du modèle deviennent rapidement visibles.

Pour les chercheurs et les professionnels souhaitant comprendre les mécanismes sous-jacents des agents offensifs avant d'adopter un outil plus complet, HackingBuddyGPT est un point de départ idéal. Sa documentation est claire, les expériences sont reproductibles et la communauté GitHub est active. Il s'intègre naturellement dans les cours de formation red team comme support pédagogique sur l'automatisation IA.

CAI — Cybersecurity AI : l'héritier de PentestGPT

CAI (Cybersecurity AI) est développé par une partie de l'équipe originale de PentestGPT, avec l'objectif explicite de dépasser les limites du prototype académique pour atteindre une utilisabilité professionnelle. Si PentestGPT était une preuve de concept remarquable, CAI est l'outil de production qui en découle logiquement.

L'architecture multi-agents de CAI est plus sophistiquée que celle de son prédécesseur : des agents spécialisés gèrent l'OSINT, le scanning réseau, les tests applicatifs et la post-exploitation, coordonnés par un orchestrateur qui maintient un graphe de connaissances sur la cible. Ce graphe permet à l'agent de relier des informations découvertes à des moments différents — une adresse email trouvée dans un fichier WHOIS et un sous-domaine identifié par DNS brute-force peuvent être mis en relation pour identifier un compte administrateur exposé.

CAI intègre également une dimension OSINT plus poussée que ses concurrents : scraping de données publiques (LinkedIn, GitHub, Shodan), corrélation d'informations de multiples sources et génération automatique d'un profil de la cible avant de commencer les tests actifs. Cette phase de reconnaissance enrichie améliore significativement la pertinence des tests ultérieurs — l'agent sait où chercher plutôt que de tester aveuglément. Pour une comparaison avec d'autres modèles d'évaluation IA, notre analyse du GPQA Diamond benchmark illustre comment les LLMs sont évalués sur des tâches complexes similaires.

En termes de performances sur XBOW, CAI se classe régulièrement parmi les trois premiers outils open source, particulièrement sur les scénarios de type « realistic enterprise » qui nécessitent une coordination multi-phases sur des durées longues.

PENTDEM : 34 outils en pipeline Docker parallèle

PENTDEM (Penetration Testing Daemon with Extended Modules) adopte une architecture orientée pipeline industriel plutôt qu'agent conversationnel. C'est un daemon — un processus qui tourne en arrière-plan et reçoit des tâches — qui orchestre 34 outils de sécurité dans des conteneurs Docker isolés, en parallèle lorsque les outils sont indépendants.

L'isolation Docker est un avantage de sécurité et de reproductibilité : chaque outil s'exécute dans un environnement propre, avec ses dépendances exactes, sans risque d'interférence entre les modules. Cette architecture est particulièrement adaptée aux pipelines CI/CD de pentest où la reproductibilité des résultats est critique — le même scan relancé deux semaines après doit produire des résultats comparables.

Le LLM d'orchestration analyse les sorties de chaque outil en parallèle, agrège les résultats et génère une priorisation des vecteurs d'attaque à explorer. Cette approche maximise la couverture de la surface d'attaque dans un temps donné, au prix d'un coût en tokens plus élevé que les architectures séquentielles. PENTDEM est particulièrement efficace pour les phases de reconnaissance et d'énumération large-scale, et moins adapté aux scénarios nécessitant un raisonnement fin sur des enchaînements d'exploitation complexes.

open·kritt : vulnérabilités réelles via workflows IA

open·kritt est l'outil le plus récent de cette liste, publié en open source début 2026, et aussi l'un des plus intéressants pour les professionnels du bug bounty. Contrairement aux autres outils qui se concentrent sur l'automatisation de techniques connues, open·kritt est conçu pour des **workflows de recherche de vulnérabilités** — trouver des bugs nouveaux dans des applications réelles, pas seulement reproduire des CVE connues.

Son architecture repose sur un graphe de décision qui guide l'agent à travers des workflows de test structurés : pour chaque type d'endpoint ou de fonctionnalité détectée, le graphe définit les hypothèses à tester et les preuves à collecter. Ce modèle est plus robuste que le ReAct libre car il contraint le raisonnement de l'agent dans un espace de décision défini par des experts humains, réduisant les hallucinations et les faux positifs.

Les résultats en bug bounty réel sont documentés : l'équipe d'open·kritt a rapporté plusieurs vulnérabilités critiques (CVSS ≥ 9.0) sur des programmes publics HackerOne en 2026, ce qui donne une crédibilité empirique que peu d'autres outils de cette liste peuvent revendiquer. La plateforme est particulièrement adaptée aux chasseurs de bugs travaillant sur des périmètres web larges où l'automatisation de la phase de triage est le goulot d'étranglement principal.

Deep-Eye, KittySploit et Strix : les émergents à surveiller

Deep-Eye est un scanner de vulnérabilités multi-IA qui se distingue par son support simultané de plusieurs backends : OpenAI, Grok, Ollama et Claude peuvent être configurés en parallèle, et l'outil agrège leurs analyses pour réduire les faux positifs. La logique est simple mais efficace : si trois modèles différents confirment indépendamment une vulnérabilité, la probabilité d'un faux positif chute drastiquement. Deep-Eye est particulièrement efficace pour la génération de payloads intelligents adaptés au contexte applicatif — il analyse la logique de l'application avant de générer des vecteurs d'injection, ce qui améliore le taux de succès sur des cibles avec des protections WAF.

KittySploit se positionne comme un successeur amélioré de Metasploit intégrant des agents IA. Avec plus de 1 150 modules d'exploitation, il combine la richesse de la base de modules Metasploit avec une couche d'orchestration LLM qui sélectionne automatiquement les modules pertinents selon la cible, adapte les paramètres et enchaîne les exploits. L'intégration est encore perfectible, mais le potentiel est considérable pour des équipes déjà familières avec l'écosystème Metasploit.

Strix adopte un modèle basé sur des graphes d'agents : différents agents spécialisés (recon, web, réseau, social engineering) sont organisés en graphe orienté, et l'orchestrateur décide dynamiquement quels agents activer selon les informations découvertes. Cette architecture distribuée est particulièrement robuste pour les engagements longs et complexes impliquant de multiples vecteurs d'attaque simultanés. Les **templates Nuclei générés par IA** méritent également une mention : plusieurs projets (dont Nuclei-AI-Templates) permettent à des LLMs de générer automatiquement des templates Nuclei pour de nouvelles CVE, démocratisant la création de détecteurs de vulnérabilités spécifiques.

Garak et Promptfoo : red-teaming des applications IA

Garak et Promptfoo occupent une niche distincte dans l'écosystème : ils ne testent pas des applications classiques, mais des **applications basées sur des LLMs**. Si vous déployez un

chatbot GPT-4 sur votre site, un agent IA pour votre service client, ou un système RAG pour l'analyse de documents internes, ces outils testent les vulnérabilités spécifiques aux LLMs : jailbreaks, prompt injection, extraction de données d'entraînement, hallucinations contrôlées et contournement des garde-fous de sécurité.

Garak est un framework de red-teaming LLM développé par NVIDIA Research. Il expose les LLMs à des centaines de scénarios d'attaque structurés — tentatives de jailbreak, prompts adversariaux, extraction d'informations privées — et quantifie leur robustesse. Le résultat est un rapport de vulnérabilités spécifique au LLM testé, avec des métriques comparatives et des recommandations de mitigation.

Promptfoo est un outil de test et d'évaluation LLM plus généraliste qui inclut des fonctionnalités de red-teaming. Il permet de définir des suites de tests automatisés pour valider le comportement d'un LLM face à des inputs adversariaux, de comparer les performances de différents modèles et de surveiller les régressions de sécurité au fil des versions. Pour les organisations qui développent des applications IA — un contexte de plus en plus courant pour les équipes de sécurité — Garak et Promptfoo sont des outils indispensables, complémentaires aux outils de pentest classiques couverts dans les sections précédentes. Notre benchmark complet des LLMs en français offre des éléments de comparaison utiles pour choisir le modèle le plus robuste dans un contexte de sécurité.

Tableau comparatif — Choisir son agent selon le contexte

Le tableau suivant synthétise les caractéristiques principales des 9 outils principaux pour faciliter le choix selon le contexte de mission.

OUTIL	TYPE	AUTONOMIE	CIBLE PRINCIPALE	LLM BACKEND	LICENCE	MATURITÉ 2026
PentestGPT	Agent ReAct	Haute (v1.0)	CTF, HTB, cibles réelles	GPT-4, Claude	MIT	⭐⭐⭐⭐ (référence)
PentesterFlow	CLI HitL	Moyenne (validée)	Web, bug bounty	OpenAI, Claude, Ollama	Open source	⭐⭐⭐⭐ (professionnel)
HexStrike AI	Serveur MCP	Dépend du LLM	Recon, scan, exploit	Tout LLM MCP-compatible	Open source	⭐⭐⭐⭐ (flexible)
pwnkit	Agent autonome	Très haute	Web apps, APIs	GPT-4o, Claude	Open source	⭐⭐⭐⭐ (XBOW top)
CyberStrike AI	Orchestrateur multi-agents	Haute	Full pentest cycle	OpenAI, Anthropic	AGPL-3.0	⭐⭐⭐⭐ (complet)
HackingBuddyGPT	Framework recherche	Moyenne	Linux privesc, AD	GPT-4, Claude, local	MIT	⭐⭐⭐ (recherche)
CAI	Multi-agents production	Haute	Full cycle + OSINT	GPT-4o, Claude	Open source	⭐⭐⭐⭐⭐ (XBOW top 3)
PENTDEM	Pipeline Docker	Haute	Recon, enum large-scale	OpenAI, Anthropic	Open source	⭐⭐⭐ (pipeline)
open·kritt	Workflows recherche vuln	Haute	Bug bounty web	OpenAI, Anthropic	Open source	⭐⭐⭐⭐ (résultats réels)

Benchmarks de performance : XBOW, HackTheBox et CTF

L'évaluation objective des agents de pentest IA a longtemps été le talon d'Achille de l'écosystème : chaque outil publiait ses propres métriques sur ses propres scénarios, rendant toute comparaison sérieuse impossible. La situation a changé avec le lancement du **benchmark XBOW** par XBOW Security en 2025. Ce benchmark standardisé évalue les agents sur un

ensemble d'environnements d'exploitation réalistes et identiques, avec des métriques uniformes : taux de réussite par catégorie, temps moyen d'exploitation, taux de faux positifs et qualité des artefacts générés.

Les résultats publiés en 2026 placent CAI, pwnkit et CyberStrike AI en tête du classement open source sur le XBOW. CAI excelle particulièrement sur les scénarios « enterprise realistic » impliquant Active Directory, où sa phase OSINT enrichie lui permet d'identifier des vecteurs d'attaque que les autres agents ratent. pwnkit domine sur les scénarios web purs — injection SQL, SSRF, IDOR — grâce à ses artefacts vérifiables et son raisonnement contextuel fin sur la logique applicative.

Sur **HackTheBox**, les résultats sont plus nuancés. PentestGPT (v1.0 agentic) résout environ 64 % des machines « Easy » et 35 % des machines « Medium » sans intervention humaine. CAI améliore ces chiffres à environ 70 % sur « Easy » et 42 % sur « Medium ». Sur les machines « Hard » et « Insane », tous les agents actuels ont un taux de réussite inférieur à 15 % — ces scénarios nécessitent une créativité offensive, une connaissance encyclopédique des techniques obscures et une persistance sur des dead-ends que les LLMs actuels n'ont pas encore.

Les **CTF (Capture The Flag)** présentent un profil différent : les challenges bien bornés avec des vecteurs d'attaque documentés sont résolus avec des taux de succès plus élevés (jusqu'à 80 % sur des CTF de niveau débutant), mais les agents peinent sur les challenges nécessitant de l'ingénierie inverse, de la stéganographie ou de la cryptanalyse — des domaines où les LLMs ont des limitations structurelles.

Une limite importante du benchmark XBOW est qu'il mesure des compétences en environnement contrôlé. En engagement réel, des facteurs comme la qualité de la documentation de scope, les restrictions réseau, les WAF, les SIEM actifs et l'interaction avec des équipes de défense influencent massivement les performances — et ne sont pas encore bien représentés dans les benchmarks existants.

Choisir son LLM backend : GPT-4o vs Claude vs Llama 3

Le choix du modèle LLM est une décision structurante qui impacte les performances, les coûts et les risques de confidentialité. Voici une analyse pragmatique des principales options en 2026.

GPT-4o (OpenAI) reste la référence en termes de performance brute sur les tâches de pentest. Son entraînement massif sur des données de sécurité — writeups CTF, rapports de vulnérabilités, documentation technique — lui confère une compréhension fine des concepts offensifs. Le principal inconvénient est le coût : un pentest web de 8 heures avec GPT-4o peut consommer entre 50 et 200 € en tokens selon la verbosité des outils et la taille des sorties traitées. La confidentialité est l'autre limite : envoyer des sorties de nmap détaillées sur une infrastructure de production à l'API OpenAI implique de transmettre des informations potentiellement sensibles à un tiers.

Claude 3.5/3.7 Sonnet (Anthropic) est souvent considéré comme légèrement supérieur à GPT-4o sur les tâches de raisonnement multi-étapes et de planification — précisément les compétences clés pour l'orchestration d'un pentest complexe. Sa fenêtre de contexte de 200K tokens est un avantage significatif pour traiter des sorties volumineuses sans troncature. Le coût est comparable à GPT-4o. Les mêmes considérations de confidentialité s'appliquent.

Llama 3 / Qwen / Mistral via Ollama résolvent les problèmes de coût et de confidentialité au prix d'une dégradation des performances. Sur les modèles 70B en local, les résultats sur des tâches de pentest sont corrects pour la reconnaissance et l'énumération, mais significativement inférieurs aux modèles cloud pour l'exploitation et le raisonnement multi-étapes. Pour les missions sur des cibles sensibles — infrastructures bancaires, données de santé, informations classifiées — l'utilisation de modèles locaux est néanmoins une obligation de sécurité opérationnelle qui prime sur la performance. Notre comparatif détaillé des LLMs disponibles en France inclut des évaluations sur des tâches techniques similaires à celles requises en pentest.

Une stratégie hybride émerge comme la meilleure pratique : utiliser un modèle cloud puissant pour la phase de planification initiale (définition de la stratégie, analyse des résultats de recon) et des modèles locaux pour l'exécution des commandes répétitives (génération de payloads, interprétation de sorties standardisées). Cette approche optimise le coût tout en limitant l'exposition d'informations sensibles.

Intégration dans un workflow de pentest professionnel

L'intégration des agents LLM dans un workflow de pentest professionnel n'est pas un remplacement mais une augmentation. Les phases où les agents apportent le plus de valeur sont la reconnaissance, l'énumération et la phase initiale de tests — des étapes longues, répétitives et bien documentées. Les phases qui bénéficient le moins de l'IA sont celles qui nécessitent une créativité offensive avancée, une connaissance intime d'architectures propriétaires ou une interaction sociale (social engineering).

Phase de reconnaissance (recon) : les agents excellent ici. Enumération DNS, scraping OSINT (WHOIS, LinkedIn, GitHub, Shodan), analyse de certificats TLS, fingerprinting applicatif — toutes ces tâches sont bien définies, leurs sorties sont structurées et les agents peuvent les enchaîner efficacement. Un agent correctement configuré peut produire en 2 heures un rapport de recon aussi complet qu'une journée de travail manuel.

Phase d'énumération : similairement efficace. Fuzzing de répertoires, énumération d'API, découverte de paramètres, test de méthodes HTTP — les agents peuvent couvrir une surface bien plus large que manuellement, avec une priorisation intelligente des résultats. La checklist de sécurité Active Directory illustre la complexité des environnements à énumérer en environnement entreprise.

Phase d'exploitation : c'est là que la supervision humaine devient critique. Les agents peuvent identifier des vecteurs d'attaque plausibles et générer des payloads initiaux, mais le taux de faux positifs est significatif. Une revue humaine systématique avant l'exécution de toute action potentiellement destructive est non-négociable en contexte professionnel.

Phase de reporting : les agents apportent une valeur significative pour la génération de rapports techniques. La structuration des preuves, la rédaction des étapes de reproduction et les recommandations de remédiation peuvent être partiellement automatisées, économisant 30 à 50 % du temps de rédaction sur un rapport complet.

Un point organisationnel important : les agents LLM génèrent des logs détaillés de toutes leurs actions et raisonnements. Ces logs constituent une traçabilité précieuse pour les rapports de pentest et pour la défense en cas de litige — l'agent a-t-il effectivement respecté le scope défini ? A-t-il exécuté des actions au-delà de ce qui était autorisé ? Pour les missions régies par un contrat de prestation PASSI, cette traçabilité est un avantage légal et qualité substantiel.

Cadre légal, éthique et responsabilités en France

L'automatisation par LLM ne modifie pas d'un iota le cadre légal applicable au pentest en France. Les agents IA sont des outils, et la responsabilité légale de leur utilisation incombe entièrement à l'opérateur humain. Ce point mérite d'être souligné sans ambiguïté.

Le **Code pénal français, article 323-1**, punit de trois ans d'emprisonnement et 45 000 € d'amende « le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données ». L'article 323-2 (atteinte au fonctionnement du système) peut s'appliquer si des tests génèrent un déni de service, même involontaire. L'article 323-3 (modification de données) couvre la modification accidentelle de données lors de tests d'exploitation. Ces infractions peuvent être commises par négligence — il suffit de dépasser le périmètre autorisé ou d'exécuter un test non prévu dans le contrat.

La **règle d'or** est simple et absolue : **autorisation écrite explicite avant tout test**, avec un périmètre (IPs, domaines, applications) clairement délimité, une fenêtre temporelle définie et une liste d'actions autorisées/interdites. Pour les engagements professionnels, un contrat de prestation signé par un représentant légal de l'organisation cible est le minimum. Pour le marché PME, l'accompagnement d'un conseil juridique pour la rédaction du contrat est fortement recommandé.

Les **programmes de bug bounty** (HackerOne, Bugcrowd, YesWeHack) définissent leur propre périmètre dans leur politique — respectez-le strictement, même quand un agent LLM explore autonomement. Configurez les agents pour qu'ils refusent d'agir sur des IPs ou des domaines hors scope, et implémentez des garde-fous techniques (liste noire d'IPs, rate-limiting des requêtes) en plus des garde-fous LLM.

Le label **PASSI (Prestataire d'Audit de Sécurité des Systèmes d'Information)** de l'ANSSI encadre les prestations de pentest pour les opérateurs d'importance vitale et les entités NIS 2. Les prestataires PASSI qui intègrent des agents LLM dans leurs missions doivent s'assurer que ces outils respectent les exigences méthodologiques PASSI, notamment en matière de traçabilité, de périmètre et de confidentialité des données. Consultez notre guide sur le mapping ReCyF ANSSI et NIS 2 pour le cadre réglementaire complet.

Limitations actuelles et perspectives 2026-2027

Après plusieurs années d'enthousiasme parfois excessif, la communauté de sécurité a développé une vision plus nuancée et réaliste des capacités et des limites des agents LLM en pentest. Voici les limitations les plus importantes à connaître avant d'intégrer ces outils dans des missions professionnelles.

Les hallucinations restent un risque opérationnel réel. Les LLMs peuvent inventer des CVE qui n'existent pas, affirmer qu'un service est vulnérable à une technique qui ne s'applique pas, ou générer des payloads syntaxiquement corrects mais inefficaces. Le taux d'hallucinations varie selon les modèles et les tâches, mais aucun modèle actuel n'est exempt de ce risque. La conséquence pratique est qu'une revue humaine systématique des conclusions de l'agent reste nécessaire avant d'inclure quelque résultat que ce soit dans un rapport de pentest professionnel.

La fenêtre de contexte est un goulot d'étranglement. Même avec les modèles à 200K tokens, un scan nmap exhaustif sur un réseau de classe B, combiné avec les sorties de gobuster sur 50 sous-domaines et les résultats d'un nikto sur 10 applications web, peut dépasser la capacité contextuelle de n'importe quel modèle. Les agents matures implémentent des mécanismes de compression et de résumé, mais ces mécanismes introduisent une perte d'information qui peut faire rater des vulnérabilités subtiles.

Le coût en tokens est un facteur limitant pour les petites équipes. Un pentest de 5 jours avec GPT-4o peut facilement consommer 300 à 500 € en coûts d'API selon l'intensité des tests et la verbosité des outils. Cette limite économique favorise les grandes équipes avec des budgets conséquents ou les organisations qui ont négocié des tarifs enterprise avec OpenAI ou Anthropic.

Les perspectives 2027 sont optimistes sur plusieurs fronts : les modèles de raisonnement (o3, Claude 4) améliorent les performances sur les tâches multi-étapes complexes ; les architectures multi-agents avec mémoire longue persistent les connaissances entre sessions ; et les benchmarks standardisés comme XBOW commencent à orienter le développement des outils vers des métriques d'utilité réelle plutôt que de performance académique. On peut raisonnablement anticiper que d'ici 2028, les agents LLM résoudront de façon autonome la majorité des machines HTB « Hard » — ce qui représentera un changement qualitatif significatif pour les équipes de pentest professionnelles.

FAQ — Agents LLM pour le Pentest

Un agent LLM peut-il remplacer un pentester humain ?

Non, pas dans l'état actuel de la technologie, et vraisemblablement pas avant plusieurs années. Les agents LLM automatisent efficacement les phases répétitives — reconnaissance, énumération, tests de vulnérabilités connues — mais peinent sur les scénarios nécessitant une créativité offensive avancée, une connaissance intime d'architectures propriétaires ou une interaction humaine (social engineering, phishing ciblé). En 2026, la valeur ajoutée des agents LLM est d'amplifier la productivité d'un pentester expérimenté, pas de le remplacer. Sur des HackTheBox « Hard » et des engagements en environnement de production complexe, le taux de réussite des meilleurs agents reste en dessous de 15 % sans intervention humaine.

Ces outils sont-ils légaux à utiliser en France ?

Les outils en eux-mêmes sont légaux — ils ne sont pas différents des scanners de vulnérabilités traditionnels (nmap, sqlmap, Metasploit) du point de vue juridique. Ce qui détermine la légalité est exclusivement l'usage : utiliser ces outils sur des systèmes pour lesquels vous avez obtenu une autorisation écrite explicite est parfaitement légal. Les utiliser sur des systèmes tiers sans autorisation constitue une infraction aux articles 323-1 et suivants du Code pénal français, passible de 3 ans d'emprisonnement et 45 000 € d'amende. L'automatisation par LLM n'atténue pas la responsabilité pénale — elle peut même l'aggraver si l'agent dépasse automatiquement le périmètre autorisé.

Quelle est la différence entre PentestGPT et CAI ?

PentestGPT est le prototype académique original, publié avec un papier à USENIX Security 2024 et optimisé pour démontrer les capacités des LLMs en pentest. CAI est son successeur orienté production, développé par une partie de la même équipe avec des objectifs différents : fiabilité, extensibilité et performances sur des engagements longs et complexes. CAI implémente une architecture multi-agents plus sophistiquée (agents spécialisés coordonnés par un orchestrateur), une phase OSINT plus complète et un meilleur système de gestion de la mémoire

entre sessions. En pratique, CAI produit de meilleurs résultats sur les scénarios d'entreprise réalistes, tandis que PentestGPT reste la référence académique pour comprendre les fondements de l'approche.

Peut-on utiliser ces agents avec des modèles locaux (Ollama/Llama) ?

Oui, plusieurs outils supportent nativement des modèles locaux via Ollama : PentesterFlow le documente explicitement, et HexStrike AI peut fonctionner avec tout LLM MCP-compatible. Les performances sont cependant significativement inférieures aux modèles cloud sur les tâches de raisonnement complexe — comptez environ 30 à 40 % de dégradation sur les benchmarks de pentest pour des modèles 70B locaux par rapport à GPT-4o. Cette dégradation est acceptable et souvent préférable lorsque la confidentialité des données de scan est une contrainte : pour des missions impliquant des infrastructures sensibles (banques, santé, défense), envoyer les sorties de nmap détaillées à une API cloud tierce est un risque opérationnel que l'option locale élimine.

Comment évaluer les performances d'un agent de pentest IA ?

Le **benchmark XBOW** est actuellement la référence pour une évaluation standardisée et comparative. Pour une évaluation interne, constituez un ensemble de cibles de test représentatives de vos contextes de mission habituels — des machines HackTheBox de niveau pertinent, des applications web vulnérables connues (DVWA, WebGoat, Juice Shop) et idéalement une mini-infrastructure simulant votre environnement type. Mesurez le taux de réussite par catégorie de vulnérabilité, le temps d'exploitation moyen et le taux de faux positifs. Évaluez également la qualité du reporting généré — un rapport de pentest utilisable directement économise plus de temps qu'un taux de détection légèrement supérieur. Enfin, testez les limites de l'agent face à des configurations complexes (WAF, authentification multi-facteur, restrictions réseau) qui sont la norme en environnement de production.

Intégrez l'IA dans vos missions de pentest

Ayi NEDJIMI Consultants intègre les agents LLM dans ses missions de pentest pour une couverture plus exhaustive de la surface d'attaque. Nos équipes combinent l'expertise humaine senior avec l'automatisation intelligente pour des rapports de qualité PASSI dans des délais réduits. Missions sur toute la France, secteurs réglementés (NIS 2, DORA, HDS).