Formation : Le Pentester Augmenté par l'IA

Modèles locaux, attaque de systèmes IA, agents autonomes

Formation pratique - 3j - 21h00 - Réf. HIA

Prix : 2100 € H.T.

NEW

Le pentest traditionnel est en mutation profonde. Les défenses s'automatisent (EDR ML, SIEM IA, WAF adaptatifs), les attaques aussi. Cecours forme des pentesters augmentés par l'IA, capables d'utiliser des modèles locaux non-censurés pour accélérer chaque phase du pentest, d'attaquer les systèmes intégrant de l'IA (chatbots, RAG, agents autonomes), et de construire leurs propres agents de pentest.L'ensemble de la formation repose sur des modèles locaux (Ollama, Dolphin-3) : zéro dépendance cloud, confidentialité mission garantie, air-gappable. API directe, pas de framework magique.L'ensemble de la formation repose sur des modèles locaux (Ollama, Dolphin-3) : zéro dépendance cloud, confidentialité mission garantie, air-gappable. API directe, pas de framework magique.

Objectifs pédagogiques

À l'issue de la formation, le participant sera en mesure de :

- Installer et configurer un environnement IA offensif local et souverain (Ollama, modèles non-censurés)
- Utiliser l'IA pour accélérer la reconnaissance, la génération de payloads et l'analyse Active Directory
- Identifier et exploiter les vulnérabilités des systèmes intégrant des LLM (OWASP Top 10 LLM 2025)
- Attaquer des architectures RAG (extraction, poisoning, injection indirecte)
- Construire un agent de pentest autonome supervisé avec boucle ReAct
- Auditer du code généré par IA (« vibe coding ») et identifier les vulnérabilités typiques

Public concerné

Pentesters, consultants sécurité offensive, analystes SOC/CERT, auditeurs sécurité, RSSI techniques souhaitant comprendre l'IA offensive et défensive.

PARTICIPANTS

Pentesters, consultants sécurité offensive, analystes SOC/CERT, auditeurs sécurité, RSSI techniques souhaitant comprendre l'IA offensive et défensive.

PRÉREQUIS

Expérience en test d'intrusion ou sécurité offensive (niveau OSCP ou équivalent recommandé).
Connaissances de base en Python (manipulation de fichiers, API REST, subprocess),Linux, Docker et les outils classiques de pentest (nmap, Burp, BloodHound). Aucune connaissance préalable en IA ou machine learning
n'est requise.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques…
Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

Prérequis

Expérience en test d'intrusion ou sécurité offensive (niveau OSCP ou équivalent recommandé). Connaissances de base en Python (manipulation de fichiers, API REST, subprocess),Linux, Docker et les outils classiques de pentest (nmap, Burp, BloodHound). Aucune connaissance préalable en IA ou machine learning n'est requise.

Modalités d'évaluation

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques…

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

Programme de la formation

1 Pourquoi le pentest tel qu'on le connaît est mort

- Comprendre l'impact des défenses IA modernes (EDR ML, NDR, SIEM IA, WAF adaptatifs) sur les techniques classiques.
- Analyser l'asymétrie attaquant/défenseur et le besoin d'augmentation par l'IA.
- Différencier modèles censurés (ChatGPT) et non-censurés (Dolphin-3) pour l'usage offensif.
- Comprendre les enjeux de souveraineté technique : confidentialité mission, air-gap, zéro télémétrie.

Travaux pratiques

Même prompt envoyé à ChatGPT (censuré) et Dolphin-3 local (non-censuré). Comparaison des résultats pour la génération de payloads offensifs.

2 Limites fondamentales des LLM

- Comprendre l'architecture LLM appliquée au pentest : tokens, context window, temperature, system prompt.
- Intégrer la limite de Shannon : le LLM compresse avec perte, il ne crée pas d'information.
- Identifier les hallucinations et leur impact (CVE inventés, commandes approximatives).
- Connaître les angles morts : vulnérabilités logiques métier, chaînes créatives, contexte environnemental, raisonnement adversarial.

3 Installation du laboratoire IA offensif

- Installer Ollama et les 4 modèles (dolphin3, qwen2.5-coder, nomic-embed-text, llama3.2).
- Créer un modèle « pentester » personnalisé avec Modelfile et system prompt offensif.
- Déployer Open WebUI pour l'interface graphique.
- Valider l'environnement et tester en mode air-gap.

Travaux pratiques

Installation complète de l'environnement IA offensif. Création du modèle custom. Validation : génération de payload et identification d'au moins une hallucination.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

• Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les formations pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
• À l'issue de chaque formation ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
• Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le participant a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Pour toute question ou besoin relatif à l'accessibilité, vous pouvez joindre notre équipe PSH par e-mail à l'adresse psh-accueil@orsys.fr.

4 Reconnaissance augmentée par IA

- Comprendre ce que le LLM apporte à la reconnaissance : synthèse, corrélation, priorisation.
- Identifier les limites : le LLM n'a pas accès au réseau, ne vérifie pas les vulnérabilités.
- Architecture d'un agent de reconnaissance : subprocess (nmap, curl) + LLM analyse.
- Utiliser la temperature adaptée : 0.3 pour l'analyse, 0.7 pour les payloads.

Travaux pratiques

Mini-agent Python de reconnaissance. Scan nmap d'une cible Docker, analyse LLM des résultats, génération automatique d'un plan d'attaque priorisé.

5 Génération de payloads et bypass de WAF

- Comparer payloads statiques (SecLists) vs payloads génératifs (LLM).
- Mettre en œuvre une boucle itérative : générer ? tester ? feedback ? adapter.
- Utiliser le feedback des réponses WAF (403, patterns bloqués) pour guider le LLM.
- Évaluer la qualité syntaxique des payloads générés et le taux de réussite.

Travaux pratiques

Fuzzing adaptatif avec LLM. Boucle automatisée de génération de payloads SQLi/XSS contre une cible avec WAF. Analyse du taux de bypass vs approche manuelle.

6 IA et Active Directory

- Comprendre le problème de l'analyse AD à grande échelle (10 000+ objets).
- Stratégie d'alimentation du LLM : pré-filtrage Python des données SharpHound/BloodHound.
- Extraction automatique : Domain Admins, Kerberoastable, AS-REP Roastable, ACLs dangereuses, délégation.
- Génération de chemins d'attaque priorisés avec commandes impacket/crackmapexec.

Travaux pratiques

Analyse de données BloodHound simulées par LLM. Pré-filtrage Python, identification de chemins d'attaque, comparaison avec l'analyse visuelle BloodHound.

7 Social engineering augmenté par IA

- Comprendre l'IA comme multiplicateur de force pour le social engineering.
- Générer des prétextes de spearphishing personnalisés à l'échelle.
- Identifier l'obsolescence des formations anti-phishing classiques face à l'IA.
- Appliquer un cadre éthique strict : red team autorisé uniquement, pas de deepfakes.

Travaux pratiques

Red team simulé contre MedTech Solutions (scénario fourni). Génération de prétextes ciblés pour 3 profils, page de phishing Office 365, exercice de détection en binôme.

8 OWASP Top 10 LLM & Agentic AI

- Maîtriser les 10 risques OWASP Top 10 for LLM Applications (2025).
- Comprendre les risques spécifiques OWASP Top 10 for Agentic AI (décembre 2025).
- Approfondir la Prompt Injection (LLM01) : injection directe, indirecte, techniques d'extraction.
- Approfondir le System Prompt Leakage (LLM07) : role-play, encodage, multi-turn, crescendo, side-channel.

Travaux pratiques

Attaque d'un chatbot vulnérable à 4 niveaux de protection croissante. Extraction de flags par prompt injection de plus en plus sophistiquée.

9 Attaque de systèmes RAG

- Comprendre l'architecture RAG : embedding, base vectorielle, recherche de similarité.
- Identifier les 4 surfaces d'attaque : poisoning, extraction, injection indirecte, manipulation de recherche.
- Comprendre pourquoi le poisoning RAG est particulièrement dangereux (influence persistante).
- Mettre en œuvre les défenses : contrôle d'accès, sanitisation des documents, détection d'anomalies.

Travaux pratiques

Exploitation d'un RAG vulnérable Docker. Extraction de document confidentiel, poisoning de la base vectorielle, injection indirecte via commentaire HTML.

10 Anatomie d'un agent IA et sa surface d'attaque

- Comprendre l'architecture agent IA : LLM Core + Tools + Memory + Planner.
- Différencier chatbot (répond) vs agent (agit) et les implications sécurité.
- Identifier les surfaces d'attaque Agentic : Tool Poisoning, Memory Manipulation, Excessive Agency, Goal Manipulation.
- Comprendre l'impact d'une prompt injection sur un agent (RCE, exfiltration, phishing).

Travaux pratiques

Exploitation de l'agent financier FinBot. Escalade de privilèges, tool abuse SQL, exfiltration de base de données, injection indirecte via email.

11 Construire un agent de pentest autonome supervisé

- Comprendre le pattern ReAct (Reasoning + Acting) pour les agents de pentest.
- Justifier le choix API directe vs frameworks (LangChain, CrewAI) : contrôle, auditabilité, stabilité.
- Implémenter le human-in-the-loop obligatoire et la blacklist de commandes dangereuses.
- Générer un résumé structuré pour le rapport de pentest.

Travaux pratiques

Construction et utilisation d'un agent de pentest supervisé. Boucle ReAct contre une cible Docker, validation humaine de chaque action, résumé final automatique.

12 Le vibe coding sous l'angle sécurité

- Comprendre les risques de sécurité du code généré par IA (patterns vulnérables dans les données d'entraînement).
- Identifier les 8 vulnérabilités typiques : SQLi, XSS, IDOR, upload non filtré, secrets hardcodés, path traversal, mots de passe en clair, absence de CSRF.
- Intégrer l'audit de code « vibe codé » dans une prestation de pentest.
- Confronter l'audit humain à l'audit par LLM du même code.

Travaux pratiques

Pentest d'une application de gestion de notes entièrement générée par IA. 8 vulnérabilités intentionnelles, 100 points. Bonus : audit du même code par le LLM et comparaison.

13 CTF « AI Red Team »

- Challenge 1 — Le chatbot bavard (30 pts) : extraction de credentials via prompt injection.
- Challenge 2 — La base empoisonnée (40 pts) : extraction de document confidentiel + poisoning RAG.
- Challenge 3 — L'agent manipulé (50 pts) : escalade de privilèges, tool abuse, exfiltration DB.
- Challenge 4 — La chaîne complète (60 pts) : web classique + prompt injection + agent = RCE.
- Challenge Bonus — Le bouclier (20 pts) : défi inversé, sécuriser un chatbot contre des attaques automatisées.

Travaux pratiques

CTF en équipes de 2-3 personnes. Infrastructure Docker complète avec scoreboard temps réel. 200 points au total. Tous les outils et techniques des jours 1 et 2 sont mobilisés.

14 Debrief et perspectives

- Positionner une prestation « AI Red Team » (argumentaire commercial, livrables, référentiels OWASP).Intégrer l'audit IA dans une méthodologie de pentest classique (cadrage, recon, exploitation, rapport).
- Comprendre les perspectives : MCP Security, agents multi-modaux, deepfakes, EU AI Act.
- Identifier les ressources de veille et d'entraînement (Gandalf, Dreadnode Crucible, PortSwigger LLM Labs, AI Goat).