

الجمهورية الشعبية الديمقراطية الجزائرية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
المدرسة العليا للإعلام الآلي - سيدي بلعباس
Higher School of Computer Science
8 Mai 1945 - Sidi Bel Abbès



Mémoire de Fin d'étude

En vue de l'obtention du diplôme d'Ingénieur d'État

Filière : Informatique

Spécialité : Ingénierie des Systèmes Informatiques (ISI)

Thème

Design and Implementation of a Hybrid Multi-Agent System for Automating Penetration Testing

Présenté par :
Haddad Ilyes

Soutenu le : **06/2026**
Devant le jury composé de

Dr. KHALDI Miloud
Pr. Rahmoun Abdelatif
M. Mahleb Ghiles
Dr. BENDELLA Mohammed Salih

Président
Encadrant
Co-Encadrant
Examineur

Année Universitaire: 2025/2026

Abstract

Penetration testing is a critical but resource-intensive security practice in which authorized testers simulate adversarial behavior to confirm the exploitability and impact of weaknesses before malicious actors can act on them. Recent surveys of LLM-driven autonomous pentesting systems report that current architectures lack a persistent and revisable representation of attack state, long-horizon planning stability, controlled and validated execution, dual-tier memory, and governance enforced at the framework level rather than the prompt level.

This thesis presents HMAS, a hybrid symbolic–neural multi-agent system designed to satisfy these five requirements simultaneously. The architecture combines a classical feasibility layer, which filters the action space through symbolic preconditions and phase ordering, with an LLM-driven selection layer that reasons over a persistent WorldState. A MasterAgent orchestrates two specialized sub-agents, ScanAgent and ExploitAgent, aligned with the phases of the Penetration Testing Execution Standard, and maintains an Attack Surface Map that gates exploitation on sufficient reconnaissance breadth. A dual-memory architecture pairs the append-only WorldState with a ChromaDB-backed retrieval-augmented knowledge base of CVE write-ups and exploit techniques, and a runtime governance layer enforces scope checks, command validation, and configurable human-in-the-loop gates before any command reaches the target.

HMAS is implemented in Python and deployed as a containerized Kali Linux agent driven through LiteLLM. It was evaluated on three tiers: 6 AUTOPENBENCH machines (17 runs), 7 Vulhub machines (19 runs), and two OverTheWire wargames. In autopilot mode, the system reaches a mean Attack Path Score of 1.000 on AUTOPENBENCH and 0.974 on Vulhub (0.986 overall across 13 targets), with an Autonomy Index of 1.0 and zero scope violations across all 36 valid benchmark runs. On the wargames it solves 28 Natas levels for \$2.39 and completes the full 33-level Bandit ladder, recovering every password through live exploitation rather than recall. These results show that combining symbolic structure, persistent shared memory, and framework-level governance with an LLM reasoning core yields an autonomous pentesting agent that is measurably more stable and reliable than the architectures surveyed, while exposing concrete limitations, in particular in pivoting within the time budget when an exploit dependency is blocked, that motivate specific directions for future work.

Keywords— autonomous penetration testing, large language models, agentic AI, multi-agent systems, hybrid symbolic–neural architecture, classical planning, memory management, rules of engagement, system evaluation, cybersecurity automation