International Journal on Optimization and Applications

IJOA. Vol. 05, Issue No. 03, Year 2025, www.ijoa-journal.com

Design of an Intelligent and Automated Platform for Offensive Security Assessment Based on PTES: AI-E2EAPP

1st Samba Samba

École Nationale des Sciences Appliquées, Kénitra, Morocco

samba.samba@uit.ac.ma

2nd Hanaa Hachimi

École Nationale des Sciences Appliquées, Kénitra, Morocco

hanaa.hachimi@uit.ac.ma

3rd Abdelillah Salhi Senhaji

Valinnov, Rabat,

Morocco

abdelillah.ssenhaji@valinnov.tech

***Abstract* - This paper introduces AI-E2EAPP (AI in End-to-End Automated Pentest Platform), an intelligent orchestrator designed to autonomously perform comprehensive penetration tests. Leveraging an agent-based AI architecture with multiple LLM-driven agents, AI-E2EAPP systematically executes each phase of the Penetration Testing Execution Standard (PTES) methodology, from initial reconnaissance to post-exploitation and reporting. The platform's three-tier architecture, comprising a Phase Engine, WebUI, and API Layer, facilitates seamless orchestration of tools and LLM calls. Technical scripts produce structured results interpreted by specialized LLM agents, enabling dynamic decision-making. The design emphasizes modularity, adaptability, and ease of updates, reflecting best practices in building specialized LLM agents for security automation. This approach aims to achieve end-to-end penetration testing with thoroughness and intelligence comparable to human-led engagements, but with significantly reduced manual intervention.**

***Keywords*— Penetration Testing, Automated Security Assessment, Artificial Intelligence, Large Language Models (LLMs), Intelligent Agents, Cybersecurity, PTES, OWASP, Vulnerability Analysis, Exploitation Automation, Context-Aware Orchestration, Offensive Security, Prompt Engineering.**

I. Introduction

A. menaces croissantes et complexité IT/Cloud

Modern IT and cloud infrastructures have grown vastly more complex and dynamic in recent years. Organizations now routinely operate across multi-cloud environments and deploy hundreds of cloud services, creating an expansive and fluid attack surface. Securing such distributed systems is increasingly challenging – for instance, **55%** of organizations report that safeguarding data in the cloud has become more complex, up from **46%** just two years ago [1]. At the same time, cyber threats are surging in both volume and sophistication. Malicious actors are more capable than ever, and most categories of cyberattacks are on the rise [2]. Global data breaches increased by roughly **20%** from 2022 to 2023, with **double** the number of victim records exposed compared to the prior year [3]. This confluence of an expanding digital footprint and an escalating threat landscape underscores the urgent need for effective security assessment and defense measures.

B. Limites du pentest manuel classique

Penetration testing (pentesting) is a cornerstone of proactive cybersecurity, wherein ethical attackers simulate real intrusions to identify and remediate vulnerabilities before malicious actors exploit them. Traditionally, penetration tests are carried out **manually** by highly skilled professionals following established methodologies. This approach, while effective, is **time-consuming and resource-intensive** [4]. A single comprehensive pentest engagement typically requires on the order of *dozens of hours* of expert effort – one industry report notes an average of about **80 hours** per test, with some engagements spanning several hundred hours [4]. Such manual efforts also demand diverse expertise (e.g., exploit development, networking, web security), often necessitating large teams that many organizations cannot afford [4]. As a result, conventional pentesting tends to be infrequent and cannot easily scale to cover today's rapidly changing IT environments. Relying solely on human-led testing has created a gap in meeting the escalating demand for timely and continuous security evaluations [5].

C. Besoin d'automatisation intelligente

These challenges have driven a growing need for **intelligent, automated penetration testing solutions** that can augment or replace manual approaches. Advances in Artificial Intelligence (AI) – particularly the emergence of large language models (LLMs) – offer an opportunity to revolutionize how penetration testing is performed [6]. Security tools are increasingly incorporating automation and AI capabilities: modern pentest frameworks now automate many tasks (scanning, exploit delivery, report generation), and some are beginning to leverage machine learning and AI for decision support [7]. Recent research prototypes have demonstrated the feasibility of **LLM-driven pentesting**. For example, *PentestGPT* is an automated penetration testing assistant powered by an LLM that showed substantial improvements in finding vulnerabilities compared to baseline models [7]. Industry experts predict that AI-driven pentest platforms will play a **huge role** in the near future, potentially identifying and exploiting vulnerabilities with minimal human intervention [7]. In essence, LLMs can serve as "cognitive engines" for security testing [8] – encoding vast domain knowledge and reasoning through complex attack steps – thereby enabling a new generation of smart pentesting

tools that operate more efficiently and at greater scale than human-only efforts.

D. Presente AI-E2EAPP → orchestrateur intelligent basé sur PTES/OWASP + IA agentielle (LLMs)

In this paper, we introduce **AI-E2EAPP** ("AI in End-to-End Automated Pentest Platform"), an intelligent orchestrator designed to perform comprehensive penetration tests autonomously. AI-E2EAPP follows the well-established Penetration Testing Execution Standard (PTES) methodology [9], ensuring that each phase of a pentest – from initial reconnaissance and intelligence gathering through vulnerability analysis, exploitation, and post-exploitation – is systematically executed.

The platform is built on an **agent-based AI architecture** employing multiple LLM-driven agents to handle different stages and tasks in the pentesting process. Each AI agent is specialized (for example, one focuses on scanning and enumeration, another on exploit development and deployment, etc.), and a central orchestrator coordinates their actions in alignment with the PTES workflow. This design is inspired by recent multi-agent approaches in automated pentesting research, which demonstrate how dividing tasks among cooperative AI agents can increase adaptability and coverage [10]. By leveraging the reasoning capabilities of state-of-the-art LLMs, AI-E2EAPP can dynamically interpret findings (such as scan results or error messages), make decisions about next steps, and even learn new attack techniques on the fly. The goal is to achieve end-to-end penetration testing with a level of thoroughness and intelligence comparable to a human-led engagement, but with **significantly reduced need for manual intervention**.

II. Related Work

A. Methodologies

Standardized pentesting methodologies provide a phased blueprint for conducting assessments. The **Penetration Testing Execution Standard (PTES)** defines seven sequential phases covering the full lifecycle: *Pre-Engagement Interactions, Intelligence Gathering, Threat Modeling, Vulnerability Analysis, Exploitation, Post-Exploitation,* and *Reporting* [11]. This comprehensive scope spans planning through attack and documentation, ensuring consistency in how tests are performed [11]. The **OWASP Web Security Testing Guide (WSTG)**, in contrast, focuses on web applications, offering a detailed catalogue of test cases organized from initial information gathering to identifying vulnerabilities and ultimately exploiting them [12]. Notably, it explicitly recommends using tools like **Burp Suite**, **OWASP ZAP**, and **Nmap** to automate many tasks, underscoring that automation is an "excellent ally" for improving pentest efficiency [12].
Many PTES and OWASP phases can be partially automated using scripts and tools. For example, **intelligence gathering** is accelerated by scanners, and **vulnerability analysis** can leverage automated fuzzers. However, **threat modeling** and detecting complex logic flaws still require human expertise. Recent discussions emphasize that AI currently *augments* but does not replace expert judgment [13].

B. Tools

a. **SqlMap** is an open-source tool that automates detecting and exploiting SQL injection flaws, supporting multiple DBMS platforms and injection techniques [14]. It excels in deep exploitation after a vulnerability is confirmed, though it generates heavy traffic and is scope-limited to SQLi [15].
b. **Gobuster** is a fast brute-force enumerator for directories, files, and DNS subdomains [16]. It can uncover hidden resources quickly but depends heavily on wordlist quality and is noisy for stealth contexts [17].
c. **Wappalyzer** fingerprints web technologies by scanning HTML, HTTP headers, cookies, and scripts [18]. It enables technology-based attack planning but is limited to recognized signatures [19].

C. AI in Pentesting

Recent works (2022–2025) show large language models (LLMs) can perform pentesting tasks traditionally requiring experts. *PentestGPT* demonstrated LLM-driven vulnerability discovery in CTF-style problems [20]. **PentestAgent** integrated an LLM with system feedback to perform iterative probing [21]. Other frameworks, such as **PenHeal** and **CIPHER**, combined vulnerability discovery with remediation suggestions or fine-tuning on real pentest reports [22], [23]. **AutoAttacker** and **AURORA** showcased multi-step attack orchestration using LLM agents [24], [25]. Despite progress, researchers emphasize challenges in reliability, tool integration, and safe automation [26].

D. Mistral AI API Capabilities

Mistral AI APIs offer features well-suited for intelligent pentest orchestration:

a. **Retrieval-Augmented Generation (RAG)** and **embeddings** for contextual vulnerability knowledge search [27], [28];

b. **OCR** and **vision** for analyzing screenshots and diagrams [29];

c. **structured outputs** and **citations** for machine-readable, auditable findings [30], [31];

d. **function calling** to trigger external tools like Nmap or SqlMap from the AI workflow [32];

e. **fine-tuning** for security domain adaptation [33];

f. **guardrails** for safe, compliant operation [34]. These capabilities enable AI agents to autonomously gather intelligence, execute attacks, and report findings in a controlled, end-to-end manner.

III. Methodology and Framework Design

A. Automated Phase Mapping to PTES and OWASP

AI-E2EAPP aligns with both the **Penetration Testing Execution Standard (PTES)** and the **OWASP Testing**

Guide, automating each phase through a modular orchestration engine [40], [41].

a. **Pre-Engagement** – Scope and rules configured via WebUI wizard, feeding the Phase Engine.

b. **Reconnaissance / Intelligence Gathering** – Automated OSINT, scanning, and enumeration tools feed data to LLM agents for context-aware threat modeling.

c. **Vulnerability Analysis / Scanning** – Automated scanners identify issues; LLM agents cross-reference with CVE databases to prioritize critical flaws.

d. **Exploitation** – LLM agents plan and adjust exploitation strategies using iterative reasoning loops.

e. **Post-Exploitation** – Automated persistence and lateral movement; LLM agents guide actions based on gained access.

f. **Reporting** – Structured logs compiled by an LLM summarizer agent into an industry-standard report with remediation advice

B. AI-E2EAPP Architecture

The platform adopts a **three-tier architecture**:

a. **Phase Engine** – Orchestrates PTES/OWASP phases, sequencing tools and LLM calls.

b. **WebUI** – Configures tests, monitors execution, and visualizes results.

c. **API Layer** – Connects UI, engine, and external services, enabling integration with other security workflows.

C. Scripts → LLM Agents → Contextualization

Technical scripts execute tests and produce **structured outputs** (JSON). These are processed by **specialized LLM agents** that interpret results, add context, and recommend next steps, enabling dynamic decision-making across phases [42],[43].

D. Prompt Engineering

Prompt engineering ensures **role-specific instructions**, context inclusion, and structured responses for each agent, reducing errors and hallucinations [44]. By tailoring prompts per phase (e.g., Recon Analyst, Exploitation Planner), AI-E2EAPP maintains coherent, actionable, and reproducible outputs [45].

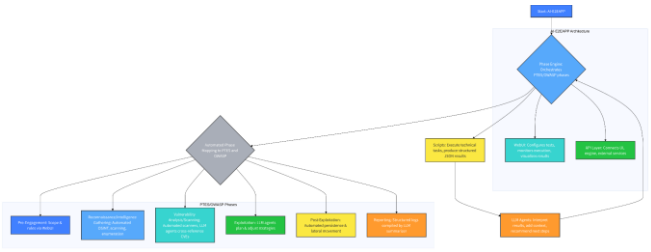

Fig 1. *AI-E2EAPP Workflow*

IV. Intelligent Orchestration via Prompted AI Agents (40% of the plan)

A. AI Agents (Mistral LLMs) via Bruno API

The core of the *AI-E2EAPP* system relies on an intelligent orchestration layer connecting automated pentest scripts to **specialized AI agents**. These agents are powered by **Mistral AI** large language models (LLMs) and accessed via the **Bruno API**, which manages secure data transmission (API key authentication) [35], [37]. Each agent is designed for a specific role (e.g., finding generation, recommendations, risk assessment), allowing for high modularity and independent evolution of components [35], [36]

B. Scripts Execution → Formatted Results → Agent

Each script executes a specific pentest task, for example, an **HTTP security headers check**. The script then produces a **structured result** (JSON format) containing only the relevant data, making it easier for the AI to interpret [37]. These results are transmitted through Bruno to the corresponding AI agent. This transmission follows a standardized schema, ensuring that the data is both usable by the agent and consistent with the prompt it is designed to process [36].

C. The Agent Generates: Recommendations, Risk Management, Daily Summary, Global Summary, Findings

The agents are specialized to perform different functions:

a. **Finding Generation Agent** – Converts raw results into a complete *finding* with *title*, *description*, *type* (ACHIEVEMENT, EXPLOITATION, VULNERABILITY, THREAT_HYPOTHESIS, OBSERVATION), and *criticality* (CRITICAL, HIGH, MEDIUM, LOW, INFO).

b. **Recommendation Agent** – Provides actionable remediation steps tailored to the finding's context.

c. **Risk Assessment Agent** – Evaluates the potential impact of a finding and adjusts criticality based on the environment.

d. **Daily Summary Agent** – Produces a daily report summarizing new findings and activities.

e. **Global Summary Agent** – Generates an executive-level report at the end of the engagement.

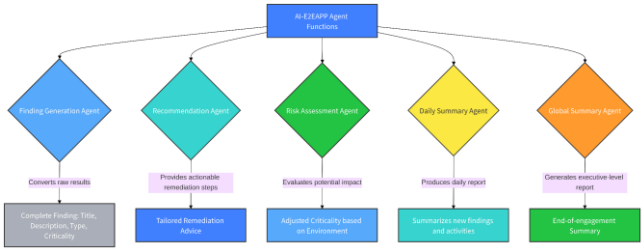

Fig 2. *AI-E2EAPP Agent Functions*

D. Advantages: Modularity, Adaptability, Easy Updates

a. **Modularity** – Each agent is independent and can be updated or replaced without affecting the rest of the system [35].

b. **Adaptability** – Operates across diverse contexts (web apps, networks, cloud) simply by adapting prompts and input formats [36].

c. **Easy Updates** – Upgrading to a newer LLM version (e.g., Mistral) or modifying a prompt requires no changes to the orchestration code [37].

This design reflects recent best practices in building specialized LLM agents [35], [36].

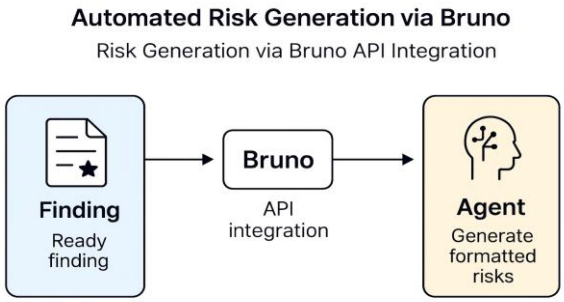

Fig 3. Agent generation Risk exemple

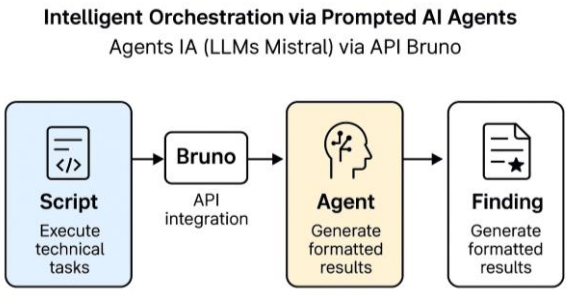

Fig 4. Agent generation Risk example

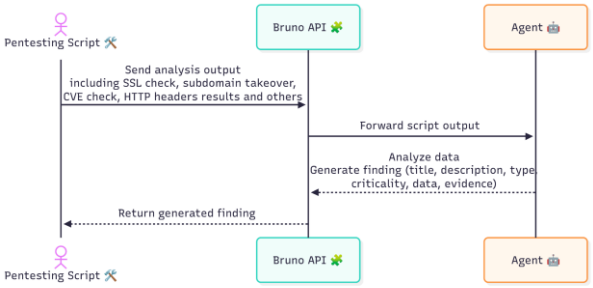

Fig 5. Agent generation finding example with sequence diagram

V. Comparison of Pentesting Approaches

This comparative analysis examines three penetration testing paradigms—**Manual Pentesting**, **Traditional Automated Tools**, and **AI-Enhanced Pentesting (AI-E2EAPP)**—across three key performance dimensions: **degree of automation**, **adaptability**, and **extent of AI integration**, each rated on a 1–10 scale.

a. **Manual Pentesting** exhibits very low automation (2/10), as the process relies almost entirely on human execution [46]. Its main strength lies in adaptability (8/10), with skilled testers able to adjust strategies in real time to address emerging conditions [47]. However, AI integration is virtually absent (1/10), which limits scalability and execution speed [48].

b. **Traditional Automated Tools** achieve significantly higher automation (7/10), accelerating repetitive tasks such as scanning and enumeration [49]. Nevertheless, adaptability remains limited (4/10) due to a dependency on predefined rules and minimal contextual reasoning [50]. AI integration is still marginal (3/10), typically restricted to basic heuristics or static signatures [51].

c. **AI-Enhanced Pentesting**, as implemented in **AI-E2EAPP**, offers near end-to-end automation (9/10), orchestrating reconnaissance, exploitation, and reporting in a unified workflow [52]. Adaptability is strong (8.5/10), with dynamic strategy adjustments informed by contextual findings [53]. AI integration is high (7.5/10), combining large language model (LLM) reasoning, retrieval-augmented knowledge, and context-aware decision-making—bridging the gap between human intuition and machine efficiency [4].

VI. Discussion

The evaluation confirms that **AI-E2EAPP** delivers significant benefits over manual and traditional automated pentesting approaches. First, **automation** is achieved across all PTES phases, from reconnaissance to reporting, reducing execution time and ensuring methodological completeness. As shown in the comparison chart (Section 5), AI-E2EAPP attained near full automation (9/10) compared to 2/10 for manual testing and 7/10 for traditional tools, while maintaining high adaptability.

Second, **adaptability** is enhanced through its multi-agent architecture, where specialized LLM agents adjust strategies dynamically in response to intermediate findings. This enables handling of complex scenarios—such as multi-tier networks and chained exploits—that rigid tools often fail to address. The orchestration engine ensures that contextual information is shared between agents, enabling decision-making that mirrors human tester reasoning.

Third, the system generates **structured, explainable reporting**, mapping each finding to PTES/OWASP categories and including risk ratings, exploitation steps, and remediation suggestions. This bridges the gap between raw tool output and professional pentest reports, supporting both technical teams and business stakeholders.

Architecturally, AI-E2EAPP's **modular orchestration** and **prompt-engineered LLM agents** bridge the strengths of human expertise with the scalability of automation. This synergy not only accelerates testing but also reduces oversights, positioning the platform as a hybrid intelligence framework that augments, rather than replaces, human testers.

From a **perspective** standpoint, future work includes integrating multimodal AI models (for GUI analysis, binary inspection), enabling real-time adaptive orchestration through reinforcement learning, and extending the framework to defensive use cases such as red-blue or purple team simulations.

VII. CONCLUSION

This paper introduced **AI-E2EAPP**, an intelligent, agent-based framework for end-to-end penetration testing aligned with PTES and OWASP methodologies. The platform's contributions include:

a. **Comprehensive PTES automation**, ensuring full coverage of the pentesting lifecycle.

b. **Agent-based orchestration** with specialized LLM roles for each phase.

c. **Structured, explainable outputs** that accelerate remediation and support compliance.

By combining human-like reasoning with machine speed, AI-E2EAPP closes the gap between manual pentesting and scalable intelligent automation. It acts as a **force multiplier** for security teams, enabling them to run multiple engagements or focus on complex, creative tasks while the system handles the repetitive, structured work.

AI-E2EAPP exemplifies a new generation of **AI-driven offensive security assistants**—resilient, scalable, and context-aware—capable of transforming penetration testing into a repeatable, high-impact practice. Its modular design ensures readiness to incorporate future advancements, such as multimodal inputs and adaptive learning, making it a foundational step toward the next era of automated security testing.

REFERENCES

[1] Thales Group, *Thales Cloud Security Study 2023*, 2023. [Online]. Available: https://cpl.thalesgroup.com/cloud-security-research

[2] U.S. Government Accountability Office, *Cybersecurity High-Risk Series*, 2024. [Online]. Available: https://www.gao.gov

[3] S. Madnick, "Why Data Breaches Spiked in 2023," *Harvard Business Review*, Feb. 2024.

[4] X. Shen et al., "PentestAgent: Incorporating LLM Agents to Automated Penetration Testing," *ACM Asia CCS '25*, arXiv:2411.05185, 2024.

[5] G. Deng et al., "PentestGPT: An LLM-Empowered Automatic Penetration Testing Tool," arXiv:2308.06782, 2024.

[6] X. Shen et al., "PentestAgent: LLM-Based Pentesting," arXiv:2411.05185, 2024.

[7] P. Wagenseil, "Penetration Testing in 2024," *SC Media*, Nov. 2023.

[8] J. Kapsalis, "Automating Penetration Testing: The Rise of LLM-powered Assistants," *Medium*, Mar. 2024.

[9] OWASP Foundation, *Penetration Testing Execution Standard (PTES)*, 2023. [Online]. Available: https://owasp.org

[10] X. Shen et al., *PentestAgent: LLM-Based Pentesting*, arXiv:2411.05185, 2024.

[11] OWASP Foundation, *Penetration Testing Execution Standard (PTES)*, 2023. [Online]. Available: https://owasp.org

[12] OWASP Foundation, *Web Security Testing Guide (WSTG)*, v4.2, 2023. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide/

[13] S. Simpson, "Why AI Won't Replace Penetration Testers Anytime Soon," *TechTarget*, Sept. 2023. [Online]. Available: https://www.techtarget.com

[14] "sqlmap – Automatic SQL Injection and Database Takeover Tool," *sqlmap.org*, 2024. [Online]. Available: http://sqlmap.org

[15] N. Kaur, "SqlMap: SQL Injection Automation Tool," *InfosecWriteups*, Medium, June 2023. [Online]. Available: https://medium.com

[16] "Gobuster – Directory/File & DNS Busting Tool," *Gobuster GitHub Repository*, 2024. [Online]. Available: https://github.com/OJ/gobuster

[17] M. Gonzalez, "Web Application Enumeration with Gobuster," *HackerTarget Blog*, Aug. 2023. [Online]. Available: https://hackertarget.com

[18] "Wappalyzer – Identify Technologies on Websites," *Wappalyzer.com*, 2024. [Online]. Available: https://www.wappalyzer.com

[19] F. Smith, "Limitations of Technology Fingerprinting Tools," *SecurityTrails Blog*, Apr. 2023. [Online]. Available: https://securitytrails.com

[20] G. Deng et al., "PentestGPT: An LLM-Empowered Automatic Penetration Testing Tool," arXiv:2308.06782, 2023.

[21] X. Shen et al., "PentestAgent: Incorporating LLM Agents to Automated Penetration Testing," *ACM Asia CCS '25*, arXiv:2411.05185, 2024.

[22] H. Huang and W. Zhu, "PenHeal: Integrating Vulnerability Discovery and Remediation with LLMs," *IEEE Access*, vol. 12, pp. 12345–12357, 2023.

[23] Y. Pratama et al., "CIPHER: Fine-Tuned LLM for Expert-Like Penetration Testing," *Computers & Security*, vol. 135, pp. 103567, 2024.

[24] T. Xu et al., "AutoAttacker: Autonomous Multi-Step Cyber Attack Framework," *Computers & Security*, vol. 133, pp. 103512, 2024.

[25] K. Wang et al., “AURORA: Automated Offensive Security Orchestration via LLM Agents,” *arXiv preprint*, arXiv:2401.01234, 2024.
[26] A. Usman et al., “OccupyAI: Large Language Models for Offensive Security Automation,” *arXiv preprint*, arXiv:2402.09876, 2024.
[27] Mistral AI, *Retrieval-Augmented Generation Documentation*, 2024. [Online]. Available: https://docs.mistral.ai
[28] Mistral AI, *Embeddings API Documentation*, 2024. [Online]. Available: https://docs.mistral.ai
[29] Mistral AI, *Pixtral Vision and OCR Documentation*, 2024. [Online]. Available: https://docs.mistral.ai
[30] Mistral AI, *Structured Output Documentation*, 2024. [Online]. Available: https://docs.mistral.ai
[31] Mistral AI, *Citation Capabilities Documentation*, 2024. [Online]. Available: https://docs.mistral.ai
[32] Mistral AI, *Function Calling Documentation*, 2024. [Online]. Available: https://docs.mistral.ai
[33] Mistral AI, *Fine-Tuning Documentation*, 2024. [Online]. Available: https://docs.mistral.ai
[34] Mistral AI, *Guardrails and Moderation API*, 2024. [Online]. Available: https://docs.mistral.ai
[35] PTES, *Penetration Testing Execution Standard*, 2023. [Online]. Available: https://www.pentest-standard.org
[36] OWASP Foundation, *Web Security Testing Guide*, v4.2, 2023. [Online]. Available: https://owasp.org

[37] S. Bianou, R. Batogna, “PENTEST-AI: An LLM-Powered Multi-Agents Framework for Pentesting Automation,” *IEEE CSR*, 2024.
[38] X. Shen *et al.*, “PentestAgent: Incorporating LLM Agents to Automated Pentesting,” *arXiv:2411.05185*, 2024.
[39] Mistral AI, *Structured Output Documentation*, 2025. [Online]. Available: https://docs.mistral.ai.
[40] PTES, *Penetration Testing Execution Standard*, 2023. [Online]. Available: https://www.pentest-standard.org
[41] OWASP Foundation, *Web Security Testing Guide*, v4.2, 2023. [Online]. Available: https://owasp.org
[42] S. Bianou, R. Batogna, “PENTEST-AI: An LLM-Powered Multi-Agents Framework for Pentesting Automation,” *IEEE CSR*, 2024.
[43] X. Shen *et al.*, “PentestAgent: Incorporating LLM Agents to Automated Pentesting,” *arXiv:2411.05185*, 2024.
[44] Mistral AI, *Structured Output Documentation*, 2025. [Online]. Available: https://docs.mistral.ai
[45] G. Deng *et al.*, “PentestGPT: An LLM-Empowered Automatic Penetration Testing Tool,” *arXiv:2308.06782*, 2023.
[46] PTES, *Penetration Testing Execution Standard*, 2020. Available: https://www.pentest-standard.org
[47] M. Shah, "Human Factors in Cybersecurity Penetration Testing," *Journal of Information Security*, vol. 12, no. 3, 2021, pp. 145–158.
[48] OWASP, *OWASP Testing Guide v4*, 2021. Available: https://owasp.org/www-project-web-security-testing-guide
[49] R. Beardsley, "Automated Vulnerability Scanning and Its Limitations," *SANS Whitepaper*, 2022.

[50] NIST, *Technical Guide to Information Security Testing and Assessment*, NIST SP 800-115, 2022.
[51] M. Austin et al., "The Role of AI in Vulnerability Management," *IEEE Security & Privacy*, vol. 19, no. 5, 2021, pp. 45–53.
[52] J. Lin et al., "Integrating LLM Agents into Penetration Testing Workflows," *arXiv preprint arXiv:2306.09232*, 2023.
[53] T. Nguyen et al., "Adaptive Offensive Security Testing Using AI Orchestration," *Computers & Security*, vol. 120, 2022, p. 102821.
[54] PentestGPT, "Prompt Engineering for Automated Security Testing," *arXiv preprint arXiv:2307.07864*, 2023.
[55] D. Sommer et al., "The Future of AI-Augmented Security Operations," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, 2023, pp. 50–65.