Оцінювання методів захисту агентних систем на основі великих мовних моделей

УДК 004.8:004.056.5

Роман Шклярський[1], Даниїл Журавчак[2]

Національний університет "Львівська політехніка",
[1]roman.shkliarskyi.asp.2025@lpnu.ua, [2]danyil.y.zhuravchak@lpnu.ua

Процес розвитку великих мовних моделей пройшов три етапи: від простих інтерфейсів завершення тексту через базові виклики API до автономних агентів з постійною пам'яттю, доступом до файлової системи та можливістю перегляду вебсторінок [1]. OWASP та NIST визначають prompt injection як основну загрозу для таких систем [2].

Архітектурна причина вразливості полягає в тому, що трансформер не розрізняє керуючі інструкції та дані користувача. Коли LLM інтегрується у виробничий конвеєр, стохастична модель поєднується з детермінованим середовищем виконання з підвищеними привілеями без механізмів перевірки походження команд [1]. Аудити 2025 року виявили вразливості в комерційних агентах для написання коду, де непрямий prompt injection призвів до виконання зловмисного коду та витоку облікових даних [3].

Атаки типу prompt injection поділяються на прямі та непрямі. Прямі передбачають формування запиту для обходу налаштувань безпеки моделі. Непрямі діють через вміст, який агент обробляє в ході роботи: вебсторінки, електронні листи або записи баз даних [2]. Успішність ручних непрямих атак становить 60–80%, атак на основі градієнтної оптимізації досягає 90–95% [4].

Системи отримання даних (RAG) вразливі до окремого класу атак. Дослідження PoisonedRAG показало, що введення п'яти шкідливих документів у базу з мільйонів записів дозволяє маніпулювати відповідями моделі з успішністю до 97% на наборі Natural Questions [5]. Показники успішності атаки за наборами даних наведено на рис. 1.

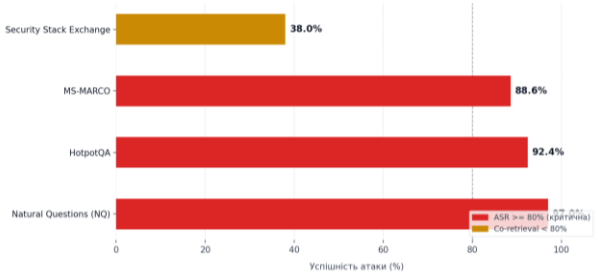

Рис.1. Успішність атаки отруєння бази знань RAG залежно від набору даних (PoisonedRAG, n = 5 документів)

Атака вимагає, щоб шкідливий текст одночасно потрапляв до топ-k результатів пошуку і містив інструкції для формування потрібної відповіді [5]. Гібридне отримання даних, що поєднує BM25 і векторний пошук, знизило

успішність атак на наборі Security Stack Exchange з 38% до 0% [6]. Архітектура RAGShield застосовує криптографічну перевірку документів перед індексуванням відповідно до NIST SP 800-53 [7].

Витік системних промптів є окремим вектором атак у багатоорендних середовищах. PROMPTPEEK показав, що спільне кешування ключ-значення дозволяє відновити вміст системного промпту іншого орендаря через аналіз часових характеристик запитів [8]. Фреймворк PLeak демонструє, що часткова реконструкція конфіденційних інструкцій можлива і без прямого доступу до інфраструктури [9]. Система StruQ розділяє вхідні дані на привілейований канал інструкцій і ненадійний канал даних, знижуючи успішність атак на 90–95% [4]. Підхід XOA забороняє моделі безпосередньо спостерігати ненадійні дані: LLM генерує сценарій, який виконується в ізольованому середовищі, а модель отримує лише кінцевий результат [10].

Фільтрація за ключовими словами та вирівнювання через RLHF не вирішують проблему на архітектурному рівні. Розмежування каналів інструкцій і даних, верифікація походження документів у RAG-конвеєрах та ізоляція середовища виконання є підходами з підтвердженою ефективністю в умовах контрольованого тестування. Стандартизація методів оцінювання захисту залишається відкритою проблемою, оскільки наявні дослідження використовують несумісні набори даних і метрики.

1. Sehwag V. et al. Clawed and Dangerous: Can We Trust Open Agentic Systems? arXiv. 2025.
2. Prompt Injection Attacks on Agentic Coding Assistants: A Systematic Analysis of Vulnerabilities in Skills, Tools, and Protocol Ecosystems. arXiv. 2025.
3. Prompt Injection Attacks in Large Language Models and AI Agent Systems. MDPI Information. 2025. Vol. 17, No. 1.
4. Chen S. et al. StruQ: Defending Against Prompt Injection with Structured Queries. USENIX Security Symposium. 2025.
5. Zou W. et al. PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models. USENIX Security Symposium. 2025.
6. Semantic Chameleon: Corpus-Dependent Poisoning Attacks and Defenses in RAG Systems. arXiv. 2025.
7. RAGShield: Provenance-Verified Defense-in-Depth Against Knowledge Base Poisoning in Government RAG Systems. arXiv. 2025.
8. PROMPTPEEK: Prompt Leakage via KV-Cache Sharing in Multi-Tenant LLM Serving. NDSS Symposium. 2025.
9. Cao Y. PLeak: Prompt Leaking Attacks against Large Language Model Applications. ACM CCS. 2024.
10. Williams D. Execute-Only Agents: Architectural Defense Against Prompt Injection for AI Agents. URL: https://people.cs.vt.edu/djwillia/papers/agenticos26-xoa.pdf (дата звернення: 05.05.2026).