УДК: 004.8 https://doi.org/10.15407/jai2026.02.177

М. С. Клименко
Державний університет інформаційно-комунікаційних технологій, Україна
7, Солом'янська вул., м. Київ, 03110
m.klymenko@duikt.edu.ua
https://orcid.org/0000-0003-4433-6641

ЗАСТОСУВАННЯ СЕМАНТИЧНОЇ МЕТРИКИ ДЛЯ ВИЯВЛЕННЯ ПРОМТ-ІН'ЄКЦІЙ У МУЛЬТИАГЕНТНИХ СИСТЕМАХ НА ОСНОВІ МОВНИХ МОДЕЛЕЙ

M. Klymenko
State University of Information and Communication Technology, Ukraine
7, Solomyanska St., Kyiv, 03110
m.klymenko@duikt.edu.ua
https://orcid.org/0000-0003-4433-6641

APPLICATION OF A SEMANTIC METRIC FOR DETECTING PROMPT INJECTION ATTACKS IN MULTI-AGENT SYSTEMS BASED ON LARGE LANGUAGE MODELS

Анотація. У роботі узагальнено сучасні дослідження останніх років з метою проведення ґрунтовного аналізу методологій виявлення, оцінки їхньої ефективності щодо еволюціонуючих зловмисних стратегій, таких як непряма ін'єкція, замасковані під предметну область шкідливі навантаження та кон'юнктивні атаки. Окреслено зміну парадигми у виявленні атак ін'єкції промтів, що полягає у переході від статичних, одноагентних механізмів захисту до динамічних, мультиагентних та структурно-орієнтованих архітектур. Новизна сучасних досліджень полягає в усвідомленні того, що традиційні лексичні та семантичні фільтри є недостатніми для протидії адаптивним зловмисникам, які використовують складні топології комунікації мультиагентних систем на основі великих мовних моделей.

Ключові слова: багатоагентні системи, відстеження механізму уваги, міркування на основі ланцюжка думок, адаптивне розподілення детекторів, доменно замасковані атаки, адаптація під час тестування, мультимодальна безпека, непряме впровадження запитів.

Abstract. The paper synthesizes contemporary research conducted in recent years to provide a comprehensive analysis of prompt injection detection methodologies and to evaluate their effectiveness against evolving adversarial strategies, including indirect prompt injection, domain-camouflaged malicious payloads, and conjunctive attacks. It outlines a paradigm shift in prompt injection attack detection, characterized by the transition from static, single-agent defense mechanisms to dynamic, multi-agent, and structurally aware architectures. The novelty of recent research lies in the recognition that traditional lexical and semantic filters are insufficient to counter adaptive adversaries that exploit the complex communication topologies of Large Language Model based multi-agent systems.

Keywords: multi-agent systems, attention mechanism tracking, chain-of-thought reasoning, adaptive detector allocation, domain-camouflaged attacks, test-time adaptation, multimodal security, indirect prompt injection.

Вступ

Стрімка інтеграція великих мовних моделей (LLM) в автономні мультиагентні системи докорінно розширила можливості штучного інтелекту, забезпечивши складну оркестрацію завдань, отримання даних у режимі реального часу та колективне розв'язання проблем у розподілених середовищах [1][2]. Однак ця архітектурна еволюція одночасно спричинила появу глибоких вразливостей безпеки, де ін'єкція підказок (prompt injection) стала домінуючим вектором загроз, здатним підривати поведінку агентів, компрометувати цілісність системи та сприяти несанкціонованим діям [3][4]. На відміну від традиційних вразливостей одноагентних систем, ін'єкція підказок у мультиагентних системах експлуатує складні протоколи комунікації, межі довіри та децентралізовані механізми міркування, притаманні цим архітектурам, створюючи поверхні атаки, які є як ширшими, так і значно складнішими для захисту [5][6].

Еволюція ландшафту загроз у мультиагентних середовищах

Атаки ін’єкції підказок еволюціонували від простих прямих перевизначень у контексті одноагентних систем до складних багатостадійних експлойтів, які використовують структурні особливості мультиагентних систем (MAS) [3][5]. У одноагентних системах зловмисники зазвичай вбудовують шкідливі інструкції безпосередньо у користувацькі запити для обходу обмежень безпеки [9]. Однак у мультиагентних середовищах модель загроз значно розширюється через залежність від міжагентної комунікації та зовнішніх джерел даних [6]. Непряма ін’єкція підказок, коли шкідливі інструкції вбудовуються у вихідні дані середовища або результати роботи інструментів, дозволяє зловмисникам захоплювати процес ухвалення рішень агентом без прямої взаємодії з користувачем [10][11]. Цей вектор особливо небезпечний у агентних робочих процесах, де LLM взаємодіють із недовіреним вебконтентом або сторонніми API, оскільки ін’єктовані інструкції інтерпретуються як легітимні дані, а не як шкідливі команди [12][13].

Складність цих загроз додатково посилюється явищем "зараження підказками" (prompt infection), коли шкідливі підказки поширюються від одного скомпрометованого агента до інших у системі, що призводить до каскадних збоїв і масштабних маніпуляцій [5]. Дослідження свідчать, що ін’єкція підказок LLM-до-LLM може відбуватися тоді, коли скомпрометований агент передає шкідливі інструкції іншому агенту, який потім виконує небезпечне навантаження, вважаючи внутрішню комунікацію довіреною [5]. Таке латеральне поширення атак підриває традиційні периметрові механізми захисту, що потребує надійних механізмів виявлення всередині системи [6]. Крім того, кон’юнктивні атаки на підказки використовують сегментацію завдань між кількома агентами, де на перший погляд нешкідливі тригери в одному агенті поєднуються з прихованими шкідливими шаблонами в іншому, активуючи небезпечну поведінку лише за виконання певних умов маршрутизації [8].

Таксономія методологій виявлення

Методології виявлення ін’єкцій підказок у мультиагентних системах можна загалом поділити на лексичні, семантичні, структурні та трансформерні підходи, кожен із яких має власні переваги та обмеження [16]. Лексичні методи покладаються на пошук шаблонів і регулярні вирази для виявлення відомих шкідливих сигнатур, однак вони часто не спрацьовують щодо перефразованих або обфускованих атак [9][17]. Семантичні методи аналізують намір і зміст вхідних даних, часто використовуючи класифікатори на основі векторних подань (embeddings) для розрізнення безпечних і шкідливих підказок [18][19]. Структурні підходи досліджують патерни уваги та внутрішні представлення LLM для виявлення аномалій, характерних для спроб ін’єкції, зокрема "ефект відволікання" (distraction effect), коли голови уваги зміщують фокус із початкових інструкцій на шкідливе навантаження [20]. Трансформерні детектори використовують донавчені моделі для класифікації вхідних даних, проте залишаються вразливими до адаптивних супротивників, які можуть обходити захист шляхом ітеративної оптимізації [17].

Семантична інваріантність наміру є важливим кроком уперед у можливостях виявлення, оскільки вона зосереджується на сталій сутності змісту, а не на поверхневих токенах [18]. Структурні методи виявлення використовують внутрішні механізми роботи LLM, зокрема механізми уваги, для ідентифікації аномальної поведінки [20]. Фреймворк Attention Tracker досліджує базові механізми ін’єкції підказок, аналізуючи, як певні голови уваги, названі "важливими головами" (important heads), зміщують свій фокус із початкових інструкцій на ін’єктований шкідливий контент [20]. Цей ефект відволікання є надійним індикатором спроб ін’єкції, дозволяючи здійснювати виявлення у реальному часі без виключної залежності від аналізу вхідного контенту

[20]. Моніторинг таких внутрішніх структурних сигналів дає змогу виявляти атаки, які успішно обходять лексичні та семантичні фільтри, забезпечуючи глибший рівень захисту [16]. Однак обчислювальні витрати на відстеження патернів уваги у великомасштабних мультиагентних системах залишаються суттєвим викликом, що потребує оптимізованих реалізацій для практичного впровадження [20].

Унікальні виклики мультиагентних систем стимулювали розроблення спеціалізованих архітектур захисту, які використовують колективну природу таких середовищ [4][23]. Мультиагентні конвеєри захисту застосовують спеціалізованих LLM-агентів у скоординованих послідовностях для виявлення та нейтралізації атак ін'єкції підказок у реальному часі [4]. Такі архітектури можуть бути організовані як послідовні ланцюги агентів або як паралельні обчислювальні вузли, кожен із яких виконує конкретну роль: санітизацію вхідних даних, перевірку намірів та валідацію вихідних даних [4][23]. Розподілена природа цих механізмів забезпечує надлишковість і стійкість, гарантує, що компрометація одного агента не призведе до компрометації всієї системи [24].

Сентинельні агенти (sentinel agents) функціонують як розподілений рівень безпеки всередині мультиагентних систем, контролюючи міжагентні комунікації та виявляючи потенційні загрози за допомогою семантичного аналізу й поведінкової аналітики [24]. Вони працюють незалежно від агентів, що виконують основні завдання, забезпечуючи неупереджену оцінку вхідних і вихідних повідомлень [24]. Завдяки інтеграції перевірки з використанням retrieval-augmented підходів та міжагентного виявлення аномалій сентинельні агенти здатні ідентифікувати невідповідності та шкідливі патерни, які окремі агенти можуть не помітити [24]. Така архітектура відповідає принципу найменших привілеїв, коли кожен агент працює з мінімально необхідними дозволами під постійним контролем, що зменшує наслідки успішних ін'єкцій [25][24].

Концепція LLM-брандмауера переносить традиційні принципи мережевої безпеки на семантичний рівень систем штучного інтелекту [26]. Агенти-валідатори виступають воротарями, здійснюючи перевірку безпеки на рівні вихідних даних, щоб гарантувати відсутність витоку інформації або виконання несанкціонованих дій [26]. Така двоагентна архітектура, що складається з основного агента-відповідача та вторинного агента-валідатора, створює надійну стратегію багаторівневого захисту [26]. Агент-валідатор аналізує відповіді основного агента відповідно до заздалегідь визначених політик безпеки та блокує будь-які відповіді, які демонструють ознаки маніпуляції або відхилення від очікуваної поведінки [26]. Цей підхід особливо ефективний у системах Retrieval-Augmented Generation (RAG), де інтеграція зовнішніх джерел знань підвищує ризик непрямих ін'єкцій [13][26].

Попри прогрес у методологіях виявлення, значні труднощі залишаються щодо впровадження цих рішень у реальних мультиагентних середовищах [16][28]. Існуючі бенчмарки часто не відображають складності та різноманітності реальних умов експлуатації, що призводить до завищених показників ефективності, які не відтворюються у виробничих системах [16][28]. Фреймворк AutoDojo демонструє, що багато запропонованих механізмів захисту є поверхневими та легко обходяться адаптивними атаками, які експлуатують межі недостатньої специфікації користувацьких запитів [28]. Крім того, оцінювання виявлення ін'єкцій підказок є сильно режимозалежним: результати суттєво варіюються між різними сценаріями виходу за межі навчального розподілу та повторними взаємодіями [16].

Статичні детектори погано відповідають динамічній моделі загроз розгорнутих LLM-систем, які після впровадження стикаються з новими типами атак і змінами розподілу даних [29]. Фреймворк EvoShield вирішує цю

проблему, формулюючи виявлення ін'єкцій підказок як задачу селективної адаптації під час тестування, поєднуючи локальний детектор на основі підказок із активними запитами до LLM для адаптації до нових векторів атак у реальному часі [29]. Такий підхід пом'якшує обмеження статичних моделей шляхом безперервного оновлення критеріїв виявлення на основі спостережуваних аномалій, хоча потребує ретельного управління обчислювальними ресурсами для уникнення надмірних затримок [29].

Традиційні бенчмарки безпеки орієнтовані на атаки та зосереджуються на їхній технічній здійсненності, ігноруючи тонкі наслідки для різних груп зацікавлених сторін [12]. Новітні підходи пропонують орієнтоване на зацікавлені сторони оцінювання для реальних вебагентів, де методи виявлення оцінюються за їхньою здатністю захищати інтереси різних користувачів і підтримувати цілісність системи в реалістичних умовах [12]. Така зміна методології оцінювання підкреслює важливість контекстно-залежних механізмів захисту, які враховують конкретні ролі та відповідальності кожного агента в системі [30][12].

Майбутнє виявлення ін'єкцій промтів у мультиагентних системах полягає в інтеграції мультимодального аналізу, формальної верифікації та проактивних механізмів стійкості [35][36]. Мультимодальні фреймворки, які включають моделі бачення та мови, здатні виявляти ін'єкції, вбудовані у зображення, метадані або повідомлення між агентами, що дозволяє протидіяти зростаючій загрозі мультимодальних ін'єкцій підказок [35][37]. Методи формальної верифікації, зокрема Metric Temporal Logic, можуть забезпечувати доведені гарантії виявлення порушень складних вимог безпеки, підвищуючи надійність систем виявлення [36].

Проактивні стратегії стійкості, включно з алгоритмами консенсусу та відмовостійкістю до візантійських збоїв (Byzantine fault tolerance), є необхідними для підтримання працездатності системи за наявності шкідливих агентів [36]. Завдяки реплікації критично важливих функцій і використанню механізмів голосування мультиагентні системи можуть приховувати збої та продовжувати коректно функціонувати навіть тоді, коли частина агентів скомпрометована [36]. Крім того, необхідним є створення комплексних фреймворків аналізу вразливостей для виявлення та усунення нових поверхонь атак, що виникають через міжагентну комунікацію та відносини довіри [6].

Висновки

Хоча ін'єкція промтів становить критичну загрозу для безпечного розгортання великих мовних моделей у мультиагентних системах, сучасні дослідження вже формують дедалі складніші стратегії виявлення та пом'якшення таких атак [3][6]. Використання мультиагентних архітектур захисту, адаптивних механізмів виявлення та суворих фреймворків оцінювання дозволяє підвищити безпеку й надійність цих складних систем штучного інтелекту [4][16][24]. Подальші інновації у сфері мультимодального аналізу, формальної верифікації та проактивної стійкості будуть необхідними для випередження еволюції зловмисних тактик і забезпечення надійної інтеграції LLM у критично важливі застосування [35][36].

Література

1. Akinrele, A., & Gowda, S. N. (2026). Prompt Injection Detection is Regime-Dependent: A Deployment-Aware Evaluation with Interpretable Structural Signals (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2605.26999
2. Arif, N. H., Lou, Q., & Zheng, M. (2026). Conjunctive Prompt Attacks in Multi-Agent LLM Systems (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2604.16543
3. Ayub, Md. A., & Majumdar, S. (2024). Embedding-based classifiers can detect prompt injection attacks (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2410.22284
4. Chang, H., Jun, Y., & Lee, H. (2025). ChatInject: Abusing Chat Templates for Prompt Injection in LLM Agents (Version 3). arXiv. https://doi.org/10.48550/ARXIV.2509.22830

5. Chen, Y., Cao, T., Li, H., Liu, Y., Li, Y., He, Y., Khoi, L. M., Song, Y., Yan, S., & Hooi, B. (2026). WebAgentGuard: A Reasoning-Driven Guard Model for Detecting Prompt Injection Attacks in Web Agents (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2604.12284
6. Du, M., Fang, H., Ma, H., Chen, J., Xu, K., Yin, Q., & Chang, E.-C. (2026). SnapGuard: Lightweight Prompt Injection Detection for Screenshot-Based Web Agents (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2604.25562
7. Ferrag, M. A., Tihanyi, N., Hamouda, D., Maglaras, L., Lakas, A., & Debbah, M. (2025). From Prompt Injections to Protocol Exploits: Threats in LLM-Powered AI Agents Workflows (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2506.23260
8. Gosmar, D., & Dahl, D. A. (2025). Sentinel Agents for Secure and Trustworthy Agentic AI in Multi-Agent Systems (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2509.14956
9. Gosmar, D., & Dahl, D. A. (2026). Prompt Injection Mitigation with Agentic AI, Nested Learning, and AI Sustainability via Semantic Caching (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2601.13186
10. Gosmar, D., Dahl, D. A., & Gosmar, D. (2025). Prompt Injection Detection and Mitigation via AI Multi-Agent NLP Frameworks (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2503.11517
11. Gulyamov, S., Gulyamov, S., Rodionov, A., Khursanov, R., Mekhmonov, K., Babaev, D., & Rakhimjonov, A. (2026). Prompt Injection Attacks in Large Language Models and AI Agent Systems: A Comprehensive Review of Vulnerabilities, Attack Vectors, and Defense Mechanisms. Information, 17(1), 54. https://doi.org/10.3390/info17010054
12. Guo, Q., Tang, J., & Huang, X. (2025). Attacking LLMs and AI Agents: Advertisement Embedding Attacks Against Large Language Models (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2508.17674
13. Hao, G., & Wu, J. (2025). Privacy-Preserving Prompt Injection Detection for Smart Cloud-Deployed Large Language Models. 2025 IEEE 10th International Conference on Smart Cloud (SmartCloud), 26–31. https://doi.org/10.1109/smartcloud66068.2025.00009
14. He, P., Xing, Y., Li, J., Dong, S., Dai, Z., Tang, X., Liu, H., Xu, H., Xiang, Z., Aggarwal, C. C., & Liu, H. (2025). Comprehensive Vulnerability Analysis is Necessary for Trustworthy LLM-MAS (Version 3). arXiv. https://doi.org/10.48550/ARXIV.2506.01245
15. Hossain, S. M. A., Shayoni, R. K., Ameen, M. R., Islam, A., Mridha, M. F., & Shin, J. (2025). A Multi-Agent LLM Defense Pipeline Against Prompt Injection Attacks (Version 4). arXiv. https://doi.org/10.48550/ARXIV.2509.14285
16. Hung, K.-H., Ko, C.-Y., Rawat, A., Chung, I.-H., Hsu, W. H., & Chen, P.-Y. (2024). Attention Tracker: Detecting Prompt Injection Attacks in LLMs (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2411.00348
17. Ju, T., Wang, Y., Hua, Y., Ma, X., Cheng, P., Zhao, H., Wang, Y., Liu, L., Xie, J., Zhang, Z., & Liu, G. (2026). Flooding spread of manipulated knowledge in LLM-based multi-agent communities. Science China Information Sciences, 69(7). https://doi.org/10.1007/s11432-024-4663-2
18. Kokkula, S., R, S., R, N., Aashishkumar, & Divya, G. (2024). Palisade -- Prompt Injection Detection Framework (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2410.21146

References

1. Akinrele, A., & Gowda, S. N. (2026). Prompt Injection Detection is Regime-Dependent: A Deployment-Aware Evaluation with Interpretable Structural Signals (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2605.26999
2. Arif, N. H., Lou, Q., & Zheng, M. (2026). Conjunctive Prompt Attacks in Multi-Agent LLM Systems (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2604.16543
3. Ayub, Md. A., & Majumdar, S. (2024). Embedding-based classifiers can detect prompt injection attacks (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2410.22284
4. Chang, H., Jun, Y., & Lee, H. (2025). ChatInject: Abusing Chat Templates for Prompt Injection in LLM Agents (Version 3). arXiv. https://doi.org/10.48550/ARXIV.2509.22830
5. Chen, Y., Cao, T., Li, H., Liu, Y., Li, Y., He, Y., Khoi, L. M., Song, Y., Yan, S., & Hooi, B. (2026). WebAgentGuard: A Reasoning-Driven Guard Model for Detecting Prompt Injection Attacks in Web Agents (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2604.12284
6. Du, M., Fang, H., Ma, H., Chen, J., Xu, K., Yin, Q., & Chang, E.-C. (2026). SnapGuard: Lightweight Prompt Injection Detection for Screenshot-Based Web Agents (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2604.25562
7. Ferrag, M. A., Tihanyi, N., Hamouda, D., Maglaras, L., Lakas, A., & Debbah, M. (2025). From Prompt Injections to Protocol Exploits: Threats in LLM-Powered AI Agents Workflows (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2506.23260
8. Gosmar, D., & Dahl, D. A. (2025). Sentinel Agents for Secure and Trustworthy Agentic AI in Multi-Agent Systems (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2509.14956
9. Gosmar, D., & Dahl, D. A. (2026). Prompt Injection Mitigation with Agentic AI, Nested Learning, and AI Sustainability via Semantic Caching (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2601.13186
10. Gosmar, D., Dahl, D. A., & Gosmar, D. (2025). Prompt Injection Detection and Mitigation via AI Multi-Agent NLP Frameworks (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2503.11517
11. Gulyamov, S., Gulyamov, S., Rodionov, A., Khursanov, R., Mekhmonov, K., Babaev, D., & Rakhimjonov, A. (2026). Prompt Injection Attacks in Large Language Models and AI Agent Systems: A Comprehensive Review of Vulnerabilities, Attack Vectors, and Defense Mechanisms. Information, 17(1), 54. https://doi.org/10.3390/info17010054

12. Guo, Q., Tang, J., & Huang, X. (2025). Attacking LLMs and AI Agents: Advertisement Embedding Attacks Against Large Language Models (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2508.17674

13. Hao, G., & Wu, J. (2025). Privacy-Preserving Prompt Injection Detection for Smart Cloud-Deployed Large Language Models. 2025 IEEE 10th International Conference on Smart Cloud (SmartCloud), 26–31. https://doi.org/10.1109/smartcloud66068.2025.00009

14. He, P., Xing, Y., Li, J., Dong, S., Dai, Z., Tang, X., Liu, H., Xu, H., Xiang, Z., Aggarwal, C. C., & Liu, H. (2025). Comprehensive Vulnerability Analysis is Necessary for Trustworthy LLM-MAS (Version 3). arXiv. https://doi.org/10.48550/ARXIV.2506.01245

15. Hossain, S. M. A., Shayoni, R. K., Ameen, M. R., Islam, A., Mridha, M. F., & Shin, J. (2025). A Multi-Agent LLM Defense Pipeline Against Prompt Injection Attacks (Version 4). arXiv. https://doi.org/10.48550/ARXIV.2509.14285

16. Hung, K.-H., Ko, C.-Y., Rawat, A., Chung, I.-H., Hsu, W. H., & Chen, P.-Y. (2024). Attention Tracker: Detecting Prompt Injection Attacks in LLMs (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2411.00348

17. Ju, T., Wang, Y., Hua, Y., Ma, X., Cheng, P., Zhao, H., Wang, Y., Liu, L., Xie, J., Zhang, Z., & Liu, G. (2026). Flooding spread of manipulated knowledge in LLM-based multi-agent communities. Science China Information Sciences, 69(7). https://doi.org/10.1007/s11432-024-4663-2

18. Kokkula, S., R, S., R, N., Aashishkumar, & Divya, G. (2024). Palisade -- Prompt Injection Detection Framework (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2410.21146

The article has been sent to the editors 05.06.26.
After processing 15.06.26.
Submitted for printing 30.06.26