Інтеграція LLM у технологію автоматизованого тестування безпеки веб-застосунків

Андрій Притула
кафедра захисту інформації
Вінницький національний технічний університет
Вінниця, Україна
andrik.pritula@gmail.com

Леонід Куперштейн
кафедра захисту інформації
Вінницький національний технічний університет
Вінниця, Україна
kupershtein.lm@gmail.com

Integration LLMs into automated web application security testing technology

Andrii Prytula
Department of information protection
Vinnytsia National Technical University
Vinnytsia, Ukraine
andrik.pritula@gmail.com

Leonid Kupershtein
Department of information protection
Vinnytsia National Technical University
Vinnytsia, Ukraine
kupershtein.lm@gmail.com

***Анотація*— Розглянуто особливості інтеграції мультимодальних великих мовних моделей у автоматизовані технології тестування безпеки веб-застосунків. Запропоновано архітектуру системи з мультимодальними агентами, визначено ключові переваги порівняно з одномодальними підходами. Проаналізовано основні ризики інтеграції та їх наслідки, окреслено заходи для мінімізації потенційних загроз. Встановлено перспективність запропонованого підходу за умови грамотного управління безпекою та контролем роботи моделей.**

***Abstract*— The paper examines the integration specifics of multimodal large language models into automated web application security testing technologies. It proposes an architecture of a multimodal agent-based system, identifying key advantages over unimodal approaches. Main integration risks and their implications are analyzed, with measures to mitigate potential threats outlined. The approach demonstrates considerable promise when properly managing security risks and ensuring effective control of model operations.**

Ключові слова— машинне навчання; тестування на проникнення; глибоке навчання; великі мовні моделі; мультиагентна система; кібербезпека;

Keywords— machine learning; penetration testing; deep learning; large language models; multi-agent system; cybersecurity;

I. Вступ

Тестування на проникнення є однією із ключових практик кібербезпеки, що дозволяє виявляти вразливості шляхом імітації реальних атак на системи. Однак проведення його вручну вимагає значних ресурсів та експертизи. Сучасні розробки у галузі штучного інтелекту пропонують нові можливості для автоматизації цих процесів [1]. Зокрема, великі мультимодальні мовні моделі (LLM) здатні генерувати текст, код і аналізувати зображення, наближаючись до рівня людського мислення у вузьких задачах. Мультимодальні LLM інтегрують різні типи даних (наприклад, текстові описи, програмний код, знімки екрана), що відкриває нові горизонти для автоматизації тестування на проникнення [2]. В рамках технологій для автоматизованого тестування на проникнення такі моделі можуть виступати інтелектуальними агентами, які планують дії, генерують експлойти або аналізують результати атак. Це дає змогу підвищити гнучкість та ефективність автоматизованого тестування на проникнення порівняно з традиційними засобами. Для прикладу, новітні моделі на кшталт GPT-4 вже продемонстрували здатність успішно виконувати складні експлойти, з якими не справлялися попередні покоління моделей [3]. Водночас інтеграція LLM у інструменти тестування на проникнення породжує питання безпеки та надійності. Необхідно оцінити, що саме змінюється при впровадженні мультимодальних LLM у архітектуру автоматизованого тестування на проникнення, для чого це робиться з точки зору вигоди та як такі зміни впливають на безпеку. У роботі розглянуто проблематику інтеграції LLM, запропоновано архітектуру технології автоматизованого тестування на проникнення з мультимодальними агентами, проведено порівняння з одномодальними моделями та проаналізовано основні ризики й наслідки впровадження.

II. Проблематика

Класичні системи автоматизації пентесту (наприклад, скрипти та спеціалізовані сканери) обмежені жорсткими

алгоритмами і правилами. Вони не здатні адаптивно реагувати на нетривіальні ситуації або творчо будувати ланцюжки атак [4]. Це створює проблему недостатньої гнучкості: багато вразливостей залишаються невиявленими, якщо вони виходять за межі закладених сценаріїв. Відтак постає потреба у впровадженні більш "розумних" компонентів, здатних розуміти контекст цілі та приймати нестандартні рішення. LLM пропонують саме такі можливості завдяки навчанням на величезних корпусах знань з кібербезпеки та програмування [5]. Інша сторона проблеми — ризики, пов’язані з самим використанням LLM у безпекових інструментах. Моделі великого масштабу можуть робити помилки або генерувати непередбачувані дії (так звані «галюцинації» моделі [6]). У контексті пентесту це означає, що агент на основі LLM може обрати неефективну або навіть небезпечну стратегію атаки. Більше того, виникають питання конфіденційності та контролю: інтеграція хмарного LLM-сервісу може вимагати відправки даних про систему-ціль сторонньому провайдеру, що потенційно створює канал витоку інформації. Таким чином, перед впровадженням LLM-агентів важливо чітко окреслити проблематику, а саме які нові загрози це приносить і як їх мінімізувати, щоб зберегти безпечність самого пентест-фреймворку. Баланс між вигодами та ризиками цієї інтеграції визначає актуальність подальших досліджень.

III. Порівняння мультимодальних і одномодальних LLM у контексті безпеки

Впровадження мультимодальної моделі порівняно з одномодальною має як переваги, так і нові виклики з точки зору безпеки. Мультимодальні LLM здатні сприймати різні види даних. Це не лише текст (описи, логи), але й зображення (скріншоти інтерфейсів, схеми мережі), програмний код чи інші структури. Натомість одномодальні LLM обмежені текстовим каналом. Це означає, що мультимодальний агент може отримати більше інформації про цільову систему. Наприклад, він проаналізує конфігураційний файл або UI-форму на зображенні і зробить висновки, недоступні чисто текстовій моделі. Це дозволить підвищити ефективність тестування безпеки, оскільки атакуючий агент з мультимодальною підтримкою може знайти більше слабких місць і обрати більш оптимальнішу техніку, порівняно з агентом, що обробляє лише текстову інформацію. Як показали дослідження, сучасна мультимодальна модель GPT-4 на практиці значно перевершує попередні текстові моделі у виконанні складних кібератак [3].

З іншого боку, мультимодальність ускладнює захист і верифікацію роботи системи [5]. Одномодальний LLM-агент може бути вразливий до ін'єкцій у підказках чи спеціально сформованих текстових входів. У випадку мультимодального з’являються аналогічні ризики в інших модальностях: шкідливе зображення (наприклад, із захованим кодом чи провокаційним текстом) може спантеличити або обдурити модель зору; спеціально підготовлений фрагмент коду може викликати непередбачувану реакцію модуля коду. Одномодальна модель простіша для аналізу та тестування, тоді як мультимодальну систему треба перевіряти на безпечність у кожному типі вводу.

Крім того, контроль над мультимодальними системами є складнішим. Одномодальні LLM можуть легше інтерпретуватися та обмежуватися правилами (наприклад, фільтрація заборонених слів у тексті). В мультимодальній же системі доводиться застосовувати набір різних політик: і для тексту, і для зображень, і для коду. Якщо текстова модель може бути навчена уникати певних небезпечних команд, то модель зору повинна додатково розпізнавати потенційно небезпечні зображення (наприклад такі, що містять QR-коди з посиланнями), а кодова модель – запобігати генерації шкідливого або некоректного коду. Таким чином, кількість векторів атак і зусиль на їх закриття зростає з розширенням модальностей. Порівняння мультимодальних і одномодальних LLM наведено у табл. 1.

ТАБЛИЦЯ І. Порівняння можливостей та ризиків одномодальних і мультимодальних LLM-агентів

Аспект	Одномодальна LLM	Мультимодальна LLM
Вхідні дані	Текст (команди, логи, описи)	Текст, зображення, код, інші типи (різні модальності)
Здатності в пентест-завданнях	Обмежений аналіз (лише текст)	Багатший контекст (бачить UI тощо)
Ризики	Ін'єкція в текстові підказки	Ін'єкція в текст, зображення, код (більше векторів)
Складність контролю	Відносно простий (єдиний канал)	Складний (потрібні політики для кожної модальності)
Ефективність автоматизації	Помірна (виконує типові сценарії)	Висока (генерує нові сценарії, адаптується)
Ймовірність помилок	Може галюцинувати текстові команди	Може помилятися в різних форматах (опис, зображення)

З погляду результату для кінцевої безпеки системи-цілі, мультимодальні LLM у тестуванні на проникнення з одного боку, покращують тестування, роблячи його ближчим до дій реальних зловмисників, що підвищує якість виявлення вразливостей. З іншого, комплексність таких агентів потенційно знижує передбачуваність, оскільки важче гарантувати, що агент не вийде за рамки заданого сценарію або не спричинить побічних ефектів. У порівнянні з одномодальними підходами, мультимодальні дають більше можливостей і вимагають більше уваги до безпечного впровадження.

IV. Архітектура інтеграції LLM у технології тестування на проникнення

Архітектура автоматизованої технології тестування на проникнення з інтеграцією LLM побудована за багатошаровим принципом агентів, кожен з яких виконує певну роль у ланцюжку атаки [7]. На верхньому рівні знаходиться центральний Агент управління, що координує роботу системи. Він отримує дані від

виконавчих агентів, ухвалює рішення про подальші кроки атаки та розподіляє завдання. На рис. 1 зображено узагальнену схему такої архітектури: керівний агент на сервері C2 взаємодіє з кількома агентами пентесту (розвідки, експлуатації, закріплення та обходу захисту).

Рис. 1. Основні компоненти мультиагентної системи для тестування на проникнення.

Основні компоненти системи включають такі виконавчі агенти [7]:

- Агент збору інформації – відповідає за первинну та поглиблену розвідку мережі й системи-цілі.
- Агент експлуатації – шукає вразливості на основі зібраних даних та запускає експлойти.
- Агент закріплення – забезпечує стійку присутність у скомпрометованій системі (бекдори, автозапуск).
- Агент обходу захисту – мінімізує виявлення атаки засобами захисту (IDS/IPS, антивірус).

Кожен з цих агентів працює напівавтономно, виконуючи специфічні етапи атаки і надсилаючи звіти до Агенту управління. Такий поділ на ролі дозволяє структуровано охопити весь цикл пентесту – від розвідки до завершального звіту – і є базою для інтеграції інтелектуальних модулів [7].

Інтеграція мультимодальних LLM здійснюється шляхом вбудовування мовних моделей у окремі задачі агентів, де потрібен «штучний інтелект». Зокрема, для Агенту управління, текстовий LLM-модуль планування, який викликається для аналізу нетипових ситуацій та генерації стратегій атаки. Цей LLM-модуль пояснює незвичні стани системи і пропонує послідовності дій, що виходять за рамки стандартних сценаріїв. Таким чином, поєднуються формальний підхід (RL-модель для ухвалення рішень) та «людське» стратегічне мислення LLM, що підвищує здатність фреймворку адаптуватися до нових середовищ без додаткового навчання.

Крім планування в керівному модулі, LLM інтегруються безпосередньо у виконання атак. Агент експлуатації використовує Code-LLM, здатний на основі опису вразливості згенерувати або адаптувати програмний код експлойта під конкретну систему. Такий підхід значно прискорює створення працездатних атак та навіть дозволяє автоматично портувати експлойти між мовами (наприклад, з Python на Rust).

Агент закріплення інтегрований з Vision-LLM: модель комп'ютерного зору аналізує знімки екрана (наприклад, GUI панелі адміністратора) і підказує, де можна непомітно впровадити бекдор. Це особливо корисно у випадках, коли відсутній прямий доступ до системи через командний рядок і потрібен аналіз візуальної інформації.

Агент обходу захисту покладається на Text-LLM для генерації обфускованих сигнатур, випадкових послідовностей пакетів і інших евристик, що ускладнюють детекцію атаки

LLM забезпечує швидке та непередбачуване варіювання технік обходу, роблячи трафік, який генерує технологія для тестування на проникнення, менш помітним. Насамкінець, варто згадати модуль звітності: після завершення тесту Text-LLM може автоматично агрегувати сирі логи та результати всіх агентів у зрозумілий підсумковий звіт (PDF/HTML) з графіками і поясненнями. Це суттєво економить час аналітика та підвищує якість звіту.

Таким чином, запропонована архітектура демонструє як мультимодальні LLM інтегруються на різних стадіях пентесту (планування, виконання, аналіз), для того – щоб підвищити інтелектуальність і покрити різні типи даних – і що вони дають в кожному випадку (прискорення, глибший аналіз, адаптивність).

V. Основні ризики та наслідки впровадження

Інтеграція LLM у пентест-фреймворк привносить ряд ризиків, які необхідно врахувати перед розгортанням такої системи. Першочерговим є ризик небажаних дій або помилкових рішень з боку LLM-агента. Через природу навчання на великих даних модель може згенерувати команду, що вийде за межі етичного пентесту (наприклад, знищення даних замість їх перегляду) або просто некоректну дію, яка не була задумана розробниками. Наслідком може стати пошкодження системи-цілі чи втрата критичної інформації під час тесту. Для мінімізації цього ризику важливо впровадити фільтрацію команд та обмеження політик. Агент менеджер повинен перевіряти кожну пропозицію LLM перед виконанням.

Другий суттєвий ризик — вразливість самої LLM до атак. Додаючи LLM до технології тестування на проникнення, відкривається новий вектор атак: зловмисник, що отримав частковий доступ до інфраструктури пентест-агента, може спробувати вплинути на LLM через його вхідні дані. Прикладом є prompt injection — коли спеціально сформований вихід цілі містить приховану команду для мовної моделі [8, 9]. Якщо модель недостатньо захищена, вона може виконати цю шкідливу команду, поставивши під загрозу увесь процес тестування [5].

Наслідки варіюються від збору неправильних даних до повного компрометування платформи. Тому захист LLM (через фільтрацію промптів, валідацію виходу моделі, ізоляцію середовища виконання) є критично важливим аспектом.

Наступний блок ризиків пов'язаний з конфіденційністю та легітимністю. Більшість передових LLM розгорнуті на сторонніх сервісах (хмарні API). Відправка даних про внутрішню мережу організації в такі сервіси під час тестування може порушувати політики

безпеки чи регуляторні вимоги. Навіть якщо модель розгорнута локально, існує ризик витоку даних через журнали чи звіти, згенеровані моделлю. Наслідком можуть стати штрафи, репутаційні втрати або непередбачене розголошення інформації про вразливості третім сторонам. Запобіжний захід – локалізація LLM (використання самостійно розгорнутих моделей) та шифрування/анонімізація чутливої інформації у взаємодії агентів.

Окремо варто згадати ризик зловживання технологією. Якщо система здатна автономно знаходити і експлуатувати вразливості, то при потраплянні до рук зловмисників вона може бути використана як потужний інструмент атаки. Дослідники вже відзначають, що атаки за допомогою GPT-4 можуть бути у кілька разів дешевшими, ніж наймання хакера-людини [3]. Наслідки впровадження такої технології на глобальному рівні – потенційне збільшення кількості автоматизованих реальних атак. Це етичний та правовий виклик, тому необхідно впроваджувати запобіжники (наприклад, вимагати авторизації і ліцензування для використання інструменту) та співпрацювати з користувачами, щоб напрацювати норми відповідального використання LLM для кібербезпеки.

Також слід врахувати наслідки для команди безпеки, що використовує такий фреймворк. З одного боку, автоматизація знижує навантаження і дозволяє фокусуватися на критичних знахідках. З іншого — вона висуває нові вимоги до кваліфікації. Фахівці повинні розуміти, як працює LLM-агент, як інтерпретувати його дії та виправляти помилки. Якщо ризики, описані вище, здійснюються, команда має бути готова швидко втрутитися в процес. Таким чином, впровадження мультимодальних LLM змінює саму суть роботи команди тестування, і ці зміни теж слід розглядати як наслідок (потреба в додатковому навчанні, оновлення протоколів реагування на інциденти тощо).

VI. Висновки

Інтеграція мультимодальних великих мовних моделей у автоматизовану технологію тестування на проникнення відкриває новий рівень можливостей для виявлення вразливостей і моделювання атак. Вона дозволяє системі виходити за межі жорстко запрограмованих сценаріїв і приймати рішення, наближені до експертних, що підвищує ефективність тестування на проникнення. У представленій архітектурі LLM-агенти успішно доповнюють кожен етап тестування – від розвідки до звітування, забезпечуючи інтелектуальний аналіз і креативність, недосяжні для традиційних методів. Втім, разом із перевагами приходять і значні виклики. Аналіз безпеки показав, що мультимодальні LLM можуть принести і вразливості, як у систему тестування на проникнення, так і у систему-ціль тестування, якщо не приділити належної уваги їхнім ризикам. Автономність моделей вимагає нових механізмів контролю. Отже, оцінка наслідків впровадження є критично важливою. LLM повинні впроваджуватись з обережністю, з прозорими обмеженнями та постійним моніторингом роботи агентів. Перші експерименти підтверджують життєздатність підходу, але для промислового застосування знадобляться додаткові дослідження з безпеки, тестування на надійність та вироблення стандартів використання. Мультимодальні LLM у тестуванні на проникнення – це перспективний напрям, який за умови грамотного керування ризиками здатен суттєво підсилити захист сучасних інформаційних систем.

Література References

[1] X. Shen *et al.* “PentestAgent: Incorporating LLM agents to automated penetration testing”. arXiv.org. [Online]. Available: https://doi.org/10.48550/arXiv.2411.05185 IEEE

[2] A. Zhuravchak та A. Piskozub, “Analysis of machine learning methods for automating penetration testing”, *Cybersecurity: Education, Science, Technique* vol. 3, no 27, pp. 54–62, 2025, doi: https://doi.org/10.28925/2663-4023.2025.27.711

[3] F. Hain, “Can autonomous LLM agents exploit one day vulnerabilities? - IONIX”, *Ionix*, March 3. 2025. [Online]. Available: https://www.ionix.io/blog/autonomous-llm-exploit-one-day-vulnerabilities-arxiv-2404-08144-explained/

[4] A. Prytula та L. Kupershtein, “Analysis of penetration testing approaches using reinforcement learning”, *Cybersecurity: Education, Science, Technique* 2025. in press.

[5] S. Guo, “Security vulnerabilities in llm-powered multi-agent systems: What developers need to know”, *Data Science Collective*, April 4. 2025. [Онлайн]. Доступно: https://medium.com/data-science-collective/security-vulnerabilities-in-llm-powered-multi-agent-systems-what-developers-need-to-know-a5a9eb4b3289

[6] S. Ray and B. Srinivasan, “Reducing hallucinations in large language models with custom intervention using Amazon Bedrock Agents | Amazon Web Services”, *AWS Machine Learning Blog*, November 26 2024. [Online]. Available: https://aws.amazon.com/blogs/machine-learning/reducing-hallucinations-in-large-language-models-with-custom-intervention-using-amazon-bedrock-agents/

[7] A. Prytula and L. Kupershtein, “Multi-agent system architecture for penetration testing”, in Vinnytsia, Ukraine, March 24–27. 2025. Vinnytsia: Vinnytsia National Technical University, 2025. [Online]. Available: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/24211

[8] J. R. Cefalu *et al.* “Evaluating the susceptibility of pre-trained language models via handcrafted adversarial examples”. September 5. 2025. arXiv.org. [Online]. Available: https://arxiv.org/abs/2209.0212

[9] V. Klysh and L. Kupershtein, “Analysis of prompt injection attacks on large language models”, in Vinnytsia, Ukraine, March 24–27. 2025. Vinnytsia: Vinnytsia National Technical University, 2025. [Online]. Available: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/24458