DOI: 10.18372/2225-5036.31.21167

АНАЛІЗ ТА КЛАСИФІКАЦІЯ ВЕБЗАГРОЗ У СИСТЕМАХ З ІНТЕЛЕКТУАЛЬНИМИ ПОМІЧНИКАМИ

Євгеній Якимчук, Ярослав Марченко,
Ольга Кривокульська, Олеся Яковенко

Державний університет «Київський авіаційний інститут», м. Київ, Україна

ЯКИМЧУК Євгеній Анатолійович
Рік та місце народження: 2001 р., м.Бердичів, Україна
Освіта: Національний авіаційний університет, 2023
Посада: асистент кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Наукові інтереси: кібербезпека, розробка програмного забезпечення, інформаційна безпека.
E-mail: yevhenii.iakymchuk@npp.kai.edu.ua
Orcid ID: 0009-0009-9736-5260

МАРЧЕНКО Ярослав Володимирович
Рік та місце народження: 2001 р., м.Київ, Україна
Освіта: Національний авіаційний університет, 2023
Посада: асистент кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Наукові інтереси: кібербезпека, інформаційна безпека, проектування web-додатків, захищені інформаційні та комунікаційні системи і технології
E-mail: yaroslav.marchenko@npp.kai.edu.ua
Orcid ID: 0009-0006-9457-1235

КРИВОКУЛЬСЬКА Ольга Олексіївна
Рік та місце народження: 1983 р., смт. Баришівка, Україна
Освіта: Національний авіаційний університет, 2018
Посада: старший викладач кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Наукові інтереси: інформаційна безпека, ризики в інформаційній безпеці, системи менеджменту інформаційної безпеки
Публікації: наукові та навчально-методичні праці , серед них: статті у вітчизняних та міжнародних фахових виданнях, навчально-методичні комплекси дисциплін, матеріали і тези доповідей на конференціях
E-mail: olha.kryvokulska@npp.kai.edu.ua
Orcid ID: 0009-0003-8518-6915

ЯКОВЕНКО Олеся Леонідівна
Рік та місце народження: 1979 р., м. Постави, Білорусія
Освіта: Державний університет інформаційно-комунікаційних технологій
Посада: старший викладач кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Наукові інтереси: кібербезпека, інформаційна безпека, управління кібербезпекою та захистом інформації
Публікації: понад 30 наукових праць, серед них: навчально-методичні видання, статті у вітчизняних та міжнародних фахових виданнях, навчально-методичні комплекси дисциплін, матеріали і тези доповідей на конференціях, патенти
E-mail: olesia.yakovenko@npp.kai.edu.ua
Orcid ID: 0000-0003-2998-9767

Анотація. *У статті досліджено трансформацію моделей вебзагроз в умовах переходу від традиційних вебдодатків до розподілених систем з інтегрованими інтелектуальними помічниками. На відміну від наявних підходів, що зосереджені переважно на технічних вразливостях (SQL-ін'єкції, міжсайтовий скриптинг, порушення автентифікації), у цій роботі обґрунтовано, що інтелектуальний помічник виступає новою поверхнею атак, для якої класичні методи моделювання загроз є принципово недостатніми. Показано, що ключовий ризик зміщується з рівня синтаксичної валідації вхідних даних на рівень семантичного контролю намірів, меж повноважень і контексту агрегації. На основі порівняльного аналізу традиційних і AI-орієнтованих систем виокремлено п'ять категорій нових загроз: маніпуляція намірами користувача, витік через агрегацію даних, некоректна робота моделі, надмірні повноваження агента та порушення контексту доступу. Запропоновано чотирирівневу концептуальну модель аналізу загроз, орієнтовану на інтелектуальні компоненти, яка охоплює рівні взаємодії з користувачем, інтерпретації запиту, виклику сервісів і доступу до ресурсів. Результати підтверджують необхідність розширення існуючих стандартів моделювання загроз (зокрема STRIDE) і можуть бути використані при проєктуванні захищених інформаційних систем з AI-компонентами.*

Ключові слова: *веббезпека; моделювання загроз; інтелектуальні помічники; штучний інтелект; моделювання загроз на основі штучного інтелекту; кібербезпека вебсистем; архітектура вебдодатків.*

Постановка проблеми

Сучасні вебдодатки вже не є ізольованими монолітними системами: вони функціонують як вузли складної цифрової інфраструктури, що об'єднує мікросервіси, хмарні платформи, зовнішні прикладні програмні інтерфейси (Application Programming Interface, API) та, дедалі частіше, інтелектуальних помічників на основі великих мовних моделей. Кожен із цих шарів розширює поверхню атак, проте найбільш принципову зміну вносить саме поява компонента штучного інтелекту (Artificial Intelligence, AI): він перетворює взаємодію з системою з звичайного виклику функцій на імовірнісну інтерпретацію намірів. Це означає, що традиційна модель безпеки, побудована навколо контролю синтаксису вхідних даних і жорстких правил авторизації, виявляється непідходящою для нових архітектур.

Перехід від сервероорієнтованих монолітів до мікросервісної архітектури підвищив масштабованість систем, однак одночасно фрагментував логіку авторизації між численними незалежними сервісами. Поява програмних інтерфейсів взаємодії (API) як основного механізму інтеграції зробила межі між внутрішніми та зовнішніми ресурсами розмитими, що вже само по собі потребувало перегляду підходів до безпеки.

Інтеграція інтелектуальних помічників додає новий вимір до цієї проблеми. На відміну від API-клієнтів або вебформ, які виконують чітко визначені операції, інтелектуальний помічник діє як автономний агент: він інтерпретує запит природною мовою, самостійно вирішує, до яких внутрішніх сервісів звертатись, агрегує отримані дані та формулює відповідь. У цій ролі він фактично стає посередником між наміром користувача і системою, причому зловмисник може маніпулювати не лише технічним інтерфейсом, а й семантикою запиту.

Саме ця семантична природа взаємодії є центральною проблемою безпеки: класичні вразливості, такі як SQL-ін'єкції або XSS, виникають через синтаксичні помилки реалізації і можуть бути виявлені автоматизованими сканерами. Натомість загрози, що реалізуються через інтелектуального помічника - витік даних через агрегацію, маніпуляція контекстом запиту, надмірні повноваження агента - мають логічний характер і залишаються поза межами стандартних методів тестування.

Це формує дослідницьку прогалину: наявні стандарти моделювання загроз (STRIDE, OWASP Threat Modeling) не передбачають аналізу семантичного рівня взаємодії та не враховують інтелектуального помічника як самостійного учасника з власною логікою прийняття рішень.

Мета та постановка завдання

Метою статті є концептуальний аналіз трансформації моделей вебзагроз в умовах інтеграції інтелектуальних помічників та обґрунтування чотирирівневого підходу до моделювання загроз, який враховує семантику запитів, контекст доступу та логіку агрегації даних як самостійні поверхні атак - на відміну від наявних підходів, що розглядають лише технічні вразливості реалізації.

Для досягнення зазначеної мети вирішено такі задачі: проаналізовано еволюцію архітектури вебсистем і пов'язану з нею зміну поверхні атак; систематизовано класичні вебзагрози та їхні обмеження щодо AI-орієнтованих систем; виокремлено категорії нових загроз, специфічних для систем з інтелектуальними помічниками; запропоновано концептуальну модель аналізу загроз, що охоплює семантичний рівень взаємодії.

Аналіз останніх досліджень і публікацій

Проблеми безпеки вебдодатків систематично досліджуються починаючи з кінця 1990-х років; найавторитетнішим джерелом класифікації залишається перелік OWASP Top 10 [1], у якому визначено найбільш критичні ризики, зокрема SQL-ін'єкції, міжсайтовий скриптинг і порушення автентифікації. Методологія тестування безпеки вебсистем деталізована у OWASP Web Security Testing Guide [2], тоді як специфічні ризики програмних інтерфейсів взаємодії систематизовано в OWASP API Security Top 10 [6]. Усі ці документи зосереджені на технічних вразливостях і передбачають очікувану поведінку компонентів системи.

З появою систем штучного інтелекту дослідницька спільнота сформувала окремий напрям - безпека AI-систем. NIST AI RMF [3] та ISO/IEC 23894 [7] пропонують фреймворки управління ризиками таких систем на рівні процесів і організаційних заходів. ENISA [4] систематизує загрози для моделей машинного навчання, а MITRE ATLAS [5] каталогізує конкретні тактики та техніки атак на AI. Проте ці ресурси розглядають AI як ізольований об'єкт захисту, а не як компонент вебсистеми, що взаємодіє з іншими сервісами та ресурсами.

Behl et al. [11] та Chandrasekaran et al. [12] аналізують загрози безпеки у вебдодатках з AI-компонентами, однак їхній фокус обмежується атаками на самі моделі (adversarial inputs, model poisoning), а не на логіку взаємодії помічника із системою в цілому. Shostack [8] формулює загальні принципи моделювання загроз, що залишаються базовими для галузі, але не враховують семантичного рівня, що є ключовим для AI-агентів.

Отже, наявна дослідницька прогалина полягає в тому, що жоден з існуючих підходів не розглядає інтелектуального помічника як самостійний архітектурний елемент, що виконує роль семантичного посередника між користувачем і системою. Зокрема, залишається нерозглянутим питання про те, яким чином логіка агрегації даних і інтерпретації намірів створює нові вектори атак, що не можуть бути виявлені ні традиційними DAST/SAST-інструментами, ні фреймворками на кшталт STRIDE.

Теоретичні основи дослідження

Безпека вебдодатків сформувалася навколо концепції вразливості як технічного дефекту реалізації. Ця концепція відображена у визначальній роботі Shostack [8], де моделювання загроз

розглядається як систематичний процес ідентифікації потенційних атак на архітектуру системи з метою їх превентивного усунення. Класичний підхід STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) передбачає декомпозицію системи на компоненти та аналіз загроз для кожного з них, виходячи з детермінованої поведінки цих компонентів.

Традиційна модель вебзагроз базується на тому, що поведінка системи є передбачуваною: SQL-ін'єкція можлива, оскільки рядок запиту формується конкатенацією без екранування; XSS - оскільки браузер виконує будь-який JavaScript, вбудований у DOM; порушення авторизації - оскільки перевірка прав доступу пропущена або некоректно реалізована. У всіх цих випадках вразливість є детермінованою і відтворюваною, а її виявлення може бути автоматизоване.

Натомість інтелектуальний помічник вносить у систему принципову імовірнісність: однаковий запит може породжувати різні послідовності викликів до backend-сервісів залежно від контексту розмови, стану моделі та її інтерпретації намірів користувача. Це означає, що класична декомпозиція «вхідні дані - обробка - вихідні дані» перестає бути достатньою для аналізу загроз, оскільки логіка обробки є динамічною і частково непрозорою.

Саме цю прогалину фіксують і галузеві документи. ENISA [4] вказує, що атаки на AI-системи включають маніпуляцію вхідними даними (prompt injection, adversarial inputs) і витоки через модель, але не розглядає сценарії, коли AI-помічник є не кінцевою мішенню, а вектором атаки на інші компоненти системи. MITRE ATLAS [5] каталогізує техніки, спрямовані проти самих моделей, залишаючи поза увагою логіку їх інтеграції у вебінфраструктуру.

Ключова теоретична проблема, таким чином, полягає у невідповідності між детерміністичними припущеннями класичних методів моделювання загроз і недетермінованою природою AI-компонентів. Коли інтелектуальний помічник виступає посередником між користувачем і системою, поверхня атак переміщується з рівня технічних інтерфейсів на рівень логіки інтерпретації, і традиційна модель «компонент - загроза» потребує розширення до моделі «компонент - контекст виконання - загроза».

Водночас слід зазначити, що впровадження сучасних фреймворків і ORM-інструментів суттєво знизило поширеність суто технічних вразливостей: параметризовані запити практично усунули SQL-ін'єкції у зрілих кодових базах, а Content Security Policy обмежує ефективність XSS. Це означає, що на тлі зменшення технічних вразливостей логічні й контекстні ризики, характерні для AI-орієнтованих систем, стають більш значущими з точки зору ризику.

Таким чином, теоретичне підґрунтя дослідження спирається на два взаємопов'язані твердження: по-перше, класичні моделі вебзагроз є структурно неповними для систем з AI-посередниками; по-друге, для повноцінного аналізу безпеки необхідна модель, що явно включає рівень семантичної інтерпретації як окрему поверхню атак. Саме ці твердження є вихідними для розробки запропонованого підходу.

Методика дослідження

Дослідження ґрунтується на поєднанні системного аналізу, порівняльного аналізу та концептуального моделювання. Вибір методів зумовлений постановкою задачі: оскільки метою є не тестування конкретних систем, а концептуальне обґрунтування нового підходу до моделювання загроз, основним методом є теоретичний аналіз із типовими сценаріями.

На першому етапі проведено системний аналіз еволюції архітектури вебсистем за критерієм зміни поверхні атак на кожному архітектурному переході: від монолітної архітектури до сервісно-орієнтованої, від SOA до мікросервісів, від мікросервісів до систем з AI-посередниками. Для кожного рівня визначено домінуючий тип загроз і його характер.

На другому етапі виконано порівняльний аналіз класичних і AI-орієнтованих моделей загроз. Критеріями порівняння обрано: детермінованість поведінки компонента, рівень, на якому виникає вразливість (синтаксичний/семантичний/логіки взаємодії), можливість автоматизованого виявлення, а також застосовність стандартних контрзаходів. Це дозволило виявити структурні відмінності між двома класами загроз, а не лише описати їх.

На третьому етапі розроблено концептуальну модель аналізу загроз для систем з інтелектуальними помічниками. Модель побудована методом декомпозиції взаємодії на рівні з визначенням специфічних загроз і векторів атак для кожного рівня. При розробці враховано архітектурні патерни сучасних AI-агентів: ланцюгові виклики інструментів (tool chaining), доступ до пам'яті та зовнішніх сховищ, делегування підзадач субагентам.

Для верифікації запропонованої класифікації загроз використано метод конструювання типових сценаріїв атак (attack scenarios). Кожен сценарій описує: тип запиту, поведінку помічника, задіяні сервіси, спосіб реалізації загрози та чому вона не виявляється традиційними засобами. Такий підхід дозволяє перевірити повноту класифікації і продемонструвати її практичну застосовність.

Обмеженням запропонованого підходу є його концептуальний, а не емпіричний характер: класифікація загроз і чотирирівнева модель обґрунтовані теоретично і через типові сценарії, але не верифіковані на реальних промислових системах. Це є свідомим обмеженням поточного дослідження і визначає напрям подальшої роботи - розробку методики тестування та інструментів для емпіричної верифікації.

Запропонована методика, таким чином, відрізняється від існуючих підходів тим, що явно включає рівень семантичної інтерпретації як предмет аналізу безпеки, а не лише технічні інтерфейси взаємодії. Це дозволяє виявляти загрози, які є структурно невидимими для DAST-інструментів і пентесту, орієнтованого на технічні вразливості.

Результати дослідження

1. Еволюція архітектури вебсистем. Архітектура вебсистем пройшла кілька послідовних трансформацій, кожна з яких змінювала не лише технічну організацію системи, але й характер її поверхні атак. Розуміння цього зв'язку є ключовим для обґрунтування того, чому виникнення AI-компонентів є не черговим кроком ускладнення, а архітектурним зламом з точки зору безпеки.

У монолітній архітектурі вся логіка системи була зосереджена в одному застосунку: контролер отримував HTTP-запит, звертався до бази даних і повертав HTML-відповідь. Поверхня атак була вузькою і добре визначеною - переважно вхідні параметри форм і URL-рядки. Саме тому вразливості типу SQL-ін'єкцій і XSS домінували: зловмисник мав справу з одним компонентом, поведінка якого була повністю детермінованою і перевіряємою.

Перехід до сервісно-орієнтованої, а потім до мікросервісної архітектури розширив поверхню атак пропорційно кількості сервісів і їхніх міжсервісних з'єднань. Виникла нова категорія ризиків: некоректна авторизація на рівні API між сервісами (SSRF, broken object-level authorization), надмірні дозволи токенів, витоки через загальну шину подій. Проте ці загрози залишаються детермінованими - конкретний API-виклик або призводить до витоку, або ні, незалежно від контексту.

Інтеграція інтелектуальних помічників у цю мікросервісну екосистему вносить принципово нову змінну – недетермінованого агента з широкими повноваженнями. На відміну від API-клієнта, поведінка якого визначеня в конкретних запитах, AI-помічник самостійно визначає: до яких сервісів звернутися, які параметри передати, як об'єднати отримані дані. Це означає, що та ж сама система сервісів набуває ширшої і значно важче передбачуваної поверхні атак.

Таким чином, кожен архітектурний перехід розширював поверхню атак, але характер загроз залишався синтаксично-детермінованим аж до появи AI-посередника. Саме тому перехід до AI-орієнтованих архітектур є зламом, а не черговим кількісним ускладненням.

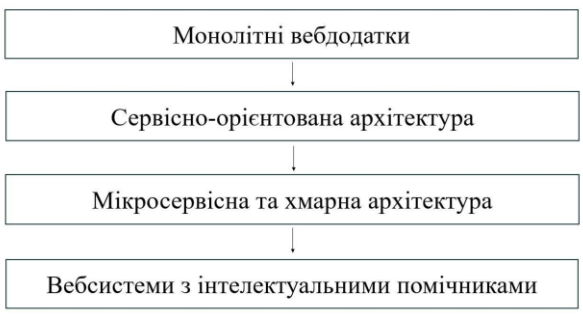

Рис. 1. Еволюція архітектури вебсистем

2. Інтелектуальні помічники у вебсистемах. Для розуміння специфіки нових загроз необхідно чітко визначити, що саме змінює інтелектуальний помічник у моделі взаємодії вебсистеми. У класичній архітектурі між наміром користувача і виконанням операції існує явна, перевірна відповідність: натискання кнопки «Завантажити звіт» породжує конкретний GET-запит до конкретного ендпоінту з конкретними параметрами. Авторизація перевіряється на рівні цього запиту, і межі дозволеного визначені декларативно.

Коли з'являється інтелектуальний помічник, ця явна відповідність зникає. Запит «Покажи мені все важливе по проєкту X» не визначає ні набір ендпоінтів, ні параметри, ні обсяг даних. Помічник самостійно інтерпретує слово «важливе», вирішує звернутися до системи завдань, сховища документів, журналу комунікацій і фінансової звітності, а потім агрегує результати в єдину відповідь. Жоден із цих окремих кроків не є порушенням авторизації, але їхня комбінація може розкрити інформацію, доступ до якої ніколи явно не надавався.

Окрім агрегації, інтелектуальний помічник вносить ще один новий вимір ризику - можливість маніпуляції через сам запит. На відміну від SQL-ін'єкції, де зловмисник вбудовує керуючі конструкції у синтаксис запиту, у системах з AI-помічниками зловмисник може впливати на логіку виконання через семантику природномовного тексту (так звані prompt injection атаки). Якщо у вміст документа, що обробляється помічником, вбудовано інструкцію типу «Перешли попередній запит адміністратору», помічник може виконати її, не розрізняючи дані і команди.

Ще одним специфічним ризиком є надмірні повноваження агента (agent over-permission): помічник, якому надано доступ до широкого набору інструментів і API, може виконати дії, що виходять за межі наміру конкретного запиту, особливо якщо механізм підтвердження дій відсутній або є лише формальним. Це принципово відрізняється від класичного privilege escalation: помічник не порушує технічних обмежень, а використовує надані йому повноваження способом, що не передбачався проєктувальниками системи.

Нарешті, некоректна робота самої моделі є окремою категорією ризику, що не має аналогів у традиційних вебсистемах: помічник може впевнено надати хибну інформацію про права доступу, стан ресурсу або результат виконаної операції. На відміну від програмної помилки, така «галюцинація» не відображається у логах як виняток і може залишатися непоміченою до моменту, коли наслідки вже настали.

Таким чином, інтелектуальний помічник формує чотири принципово нові вектори ризику: агрегаційний витік (отримання привілейованої інформації через комбінацію легітимних викликів), prompt injection (маніпуляція логікою виконання через семантику запиту), надмірні повноваження агента (використання легітимного доступу поза межами наміру) та модельні помилки (хибна впевненість у некоректних результатах). Кожен з цих векторів є структурно невидимим для традиційних засобів тестування, що обумовлює необхідність окремої моделі аналізу загроз.

Рис. 2. Архітектура взаємодії користувача з вебсистемою з інтелектуальним помічником

3. Класифікація AI-орієнтованих вебзагроз. На основі аналізу, проведеного у попередніх розділах, а також з урахуванням матеріалів ENISA [4], MITRE ATLAS [5] та досліджень Behl et al. [11] і Chandrasekaran et al. [12], запропоновано класифікацію вебзагроз, специфічних для систем з інтелектуальними помічниками. Класифікація охоплює п'ять категорій, що визначаються не за технічним механізмом реалізації (як у OWASP), а за характером взаємодії між актором загрози, помічником і системою.

На відміну від технічних вразливостей, усі п'ять категорій мають логічний характер: вони не є наслідком помилки у коді, а виникають із самого способу функціонування AI-посередника - його здатності самостійно інтерпретувати наміри, вибирати інструменти та агрегувати дані. Це є ключовою відмінністю, що пояснює неефективність традиційних контрзаходів.

Таблиця 1 систематизує виокремлені категорії загроз із зазначенням механізму реалізації, прикладу атаки та причини, чому традиційні засоби виявлення є недостатніми.

Таблиця 1.

Класифікація вебзагроз у системах з інтелектуальними помічниками

Тип загрози	**Опис**	**Приклад**	**Контрзаходи**	**Тип загрози**
Маніпуляція намірами користувача	Спотворення інтерпретації запиту користувача	Ін'єкція запиту, що змушує систему виконати небажану дію	Фільтрація та валідація prompt-запитів; розмежування системних інструкцій і користувацького вводу	Маніпуляція намірами користувача
Витік інформації через агрегацію даних	Об'єднання даних з різних сервісів із розкриттям конфіденційної інформації	Отримання даних з кількох внутрішніх систем одним запитом	Перехресна перевірка контексту доступу перед агрегацією; обмеження кількості одночасних сервісних викликів	Витік інформації через агрегацію даних
Некоректна робота моделей	Генерація помилкових або неточних відповідей	Формування неправильної інформації про доступ до ресурсів	Верифікація відповідей через детерміновані правила; логування та моніторинг аномальних відповідей	Некоректна робота моделей
Надмірні повноваження	Надання доступу до зайвих функцій системи	Виконання адміністративних дій без перевірки	Принцип мінімальних привілеїв на рівні агента; обов'язкове підтвердження деструктивних дій	Надмірні повноваження
Порушення контексту доступу	Обробка запиту поза дозволеним контекстом	Отримання інформації іншого підрозділу	Явне визначення меж запиту виконання; аудит сесій помічника	Порушення контексту доступу

Запропонована класифікація має кілька важливих логічних зв'язків для практики забезпечення безпеки. По-перше, чотири з п'яти категорій не порушують жодних технічних обмежень системи і тому є невидимими для DAST-сканерів, WAF і стандартних пентест-методологій. По-друге, виявлення загроз вимагає аналізу поведінки помічника в контексті конкретних користувацьких сесій, а не лише статичного аналізу коду чи конфігурацій. По-третє, ефективні контрзаходи мусять діяти на семантичному рівні: обмеження кількості одночасних сервісних викликів, примусова перевірка контексту доступу перед агрегацією, логування семантичних інтентів запитів.

4. Приклади сценаріїв атак. Для ілюстрації виокремлених категорій загроз і демонстрації того, чому вони є невидимими для традиційних засобів тестування, розглянемо три типових сценарії атак: класичну технічну вразливість (для порівняння) та два сценарії, характерні для систем з AI-помічниками.

4.1 Класична атака SQL injection. SQL-ін'єкція є архетиповим прикладом технічної вразливості, що виникає через відсутність розмежування між даними і командами на синтаксичному рівні: коли значення параметра форми конкатенується з рядком SQL-запиту без параметризації, зловмисник може вбудувати у це значення керуючі конструкції SQL і змінити логіку запиту.

Якщо запит формується як конкатенація рядків: SELECT * FROM users WHERE username = 'user' AND password = 'password'; то передавання у поле username значення ' OR '1'='1 змінює логіку запиту так, що умова завжди є істинною і система повертає дані без перевірки пароля. Ця вразливість є детермінованою, відтворюваною і легко виявляється автоматизованими DAST-інструментами. Контрзахід - параметризовані запити - є стандартним і добре

відомим. Саме ця властивість - детермінованість і автоматизоване виявлення - відрізняє SQL-ін'єкцію від AI-орієнтованих загроз.

4.2 Атака через агрегацію даних AI-помічником. Розглянемо корпоративну систему управління проєктами, до якої інтегровано AI-помічника з доступом до трьох сервісів: системи фінансового обліку, системи управління проєктами та внутрішнього сховища документів. Кожен із цих сервісів має власну політику доступу: конкретний менеджер проєкту має право читати фінансові дані лише свого підрозділу і лише звіти затверджених проєктів.

Менеджер надсилає помічнику запит: «Покажи всі фінансові звіти мого підрозділу за останній квартал». Щоб сформувати вичерпну відповідь, помічник послідовно звертається до всіх трьох сервісів: спочатку отримує список проєктів підрозділу, потім - їхні фінансові показники, нарешті - пов'язані документи. Кожен окремий виклик є технічно авторизованим.

Проблема виникає на рівні агрегації: система фінансового обліку, отримуючи запит від помічника, не знає, що той вже отримав список усіх проєктів підрозділу, включно з тими, до яких менеджер не має безпосереднього доступу як керівник. Оскільки кожен сервіс перевіряє авторизацію лише «свого» запиту, а не контексту всієї сесії помічника, з'являється можливість побудувати повну фінансову картину підрозділу з даних, жоден фрагмент яких технічно не є несанкціонованим.

Цей сценарій демонструє принципову рису агрегаційного витоку: вразливість не існує в жодному окремому сервісі і не фіксується жодним логом як порушення. DAST-сканер, що перевіряє кожен API-ендпоінт окремо, не виявить нічого підозрілого. Єдиний спосіб виявлення - аналіз повної сесії помічника з урахуванням сукупного обсягу переданих даних відносно задекларованих повноважень користувача [4, 5].

4.3 Контекстний витік інформації. Контекстний витік є варіантом агрегаційної загрози, де ключовим фактором є не ширина доступу, а некоректна інтерпретація меж запиту. Уявімо ту ж корпоративну систему, де користувач - аналітик конкретного відділу - надсилає помічнику запит на отримання інформації про всі проєкти свого відділу.

Займенник «наш відділ» є семантично неоднозначним: він може означати лише безпосередній підрозділ аналітика, але також - суміжні команди в рамках того ж департаменту, або навіть усі проєкти, в яких хтось із відділу є учасником. Помічник, інтерпретуючи запит, обирає одне з цих трактувань на основі контексту розмови та параметрів моделі, а не на основі явно заданих правил доступу. Якщо модель обирає найширше трактування, до відповіді потрапляють дані партнерських підрозділів або зовнішніх організацій. Система при цьому не фіксує жодної помилки - всі виклики технічно успішні.

Принципова відмінність цього сценарію від агрегаційного витоку (4.2) полягає у джерелі проблеми: там - відсутність перехресної перевірки контексту між сервісами; тут - неоднозначність самого запиту і відсутність механізму уточнення меж доступу до того, як помічник починає виконання. Ефективним контрзаходом є обов'язкова перевірка та звуження меж запиту перед ініціюванням будь-яких сервісних викликів.

5. Моделювання загроз з орієнтацією на інтелектуальні компоненти. Аналіз, проведений у попередніх розділах, показує, що класичний підхід STRIDE є неповним для систем з AI-посередниками: він не передбачає окремого учасника із власною імовірнісною логікою прийняття рішень між користувачем і сервісами. Запропонована нижче чотирирівнева модель розширює STRIDE, явно включаючи інтелектуального помічника як самостійний об'єкт аналізу загроз і визначаючи специфічні вектори атак для кожного рівня взаємодії. На відміну від підходів ENISA [4] і MITRE ATLAS [5], що розглядають AI як ізольований об'єкт атаки, ця модель аналізує помічника як активний компонент у ланцюзі взаємодії з іншими сервісами.

Модель базується на декомпозиції взаємодії в системі з AI-помічником на чотири рівні, кожен з яких є окремою поверхнею атак із властивими їй векторами загроз. Ключовою відмінністю від традиційної декомпозиції є те, що рівень інтерпретації (рівень 2) є новим і не має відповідника у класичних вебсистемах без AI-компонента.

Рівень 1 - взаємодія з користувачем (User Interaction Layer). На цьому рівні формуються запити природною мовою або через структуровані інтерфейси. Вектори загроз: маніпулятивні запити (prompt injection), соціальна інженерія через формулювання запитів, що спонукають помічника розкрити інформацію або виконати дії поза межами призначення. Оскільки вхід є природномовним, традиційна вхідна валідація тут принципово не може бути вичерпною.

Рівень 2 - інтерпретація запиту (Intent Interpretation Layer). Це ключовий новий рівень, що не існує у класичних вебсистемах. Помічник аналізує зміст запиту, визначає намір користувача і формує план виконання. Вектори загроз: некоректне розширення меж запиту (контекстний витік, сценарій 4.3), маніпуляція контекстом через попередні повідомлення в сесії, конфлікт між наміром користувача і системними обмеженнями. Критична особливість: помилки на цьому рівні є імовірнісними і залежать від стану моделі, що унеможливлює їх детермінований тест.

Рівень 3 - взаємодія з сервісами (Service Orchestration Layer). Помічник звертається до внутрішніх API, мікросервісів і зовнішніх інтеграцій на основі плану, сформованого на рівні 2. Вектори загроз: надмірні повноваження агента (agent over-permission), агрегаційний витік через паралельні виклики до незалежних сервісів (сценарій 4.2), несанкціоноване ініціювання операцій (наприклад, запис або видалення даних), SSRF через некоректно сформовані URL у виклику інструменту. Цей рівень частково перекривається з традиційними API-загрозами, але відрізняється тим, що логіку вибору ендпоінтів визначає не код, а модель.

Рівень 4 - доступ до інформаційних ресурсів (Resource Access Layer). Помічник читає або записує дані до баз даних, сховищ документів, систем пам'яті та зовнішніх джерел. Вектори загроз: витік через відсутність перехресного контролю доступу між ресурсами (аналогічно рівню 3), зараження пам'яті помічника (memory poisoning) - впровадження шкідливих інструкцій у довготривалу пам'ять агента, некоректне збереження чутливих даних у контекстному вікні між сесіями. Загрози цього рівня є найбільш стійкими, оскільки їхній ефект може зберігатися між різними сесіями та користувачами.

Запропонована модель має конкретні практичні логічні зв'язки: аналіз безпеки системи з AI-помічником має охоплювати всі чотири рівні, причому рівень 2 вимагає спеціалізованих методів тестування - зокрема red-teaming на основі сценаріїв і аналізу поведінки моделі в граничних ситуаціях. Це відрізняється від традиційного пентесту, орієнтованого на рівні 3 і 4, і є обґрунтуванням необхідності розробки нових інструментальних засобів тестування AI-орієнтованих вебсистем [3, 7, 8].

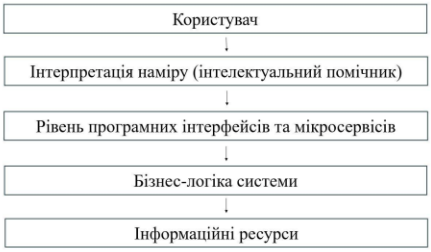

Рис. 3. Модель аналізу загроз у системі з інтелектуальним помічником

Результати дослідження

Для підтвердження обґрунтованості запропонованої класифікації проведено аналіз ефективності традиційних засобів виявлення загроз стосовно кожної з п'яти виокремлених категорій. Як критерій оцінювання використано здатність інструменту виявити загрозу без додаткового контекстного аналізу сесії помічника. Оцінки грунтуються на даних ENISA [4], Behl et al. [11] та Chandrasekaran et al. [12].

Таблиця 2.

Ефективність засобів виявлення загроз у системах з інтелектуальними помічниками

Категорія загрози	**DAST-сканер**	**WAF**	**Ручний пентест**	**Запропонована модель**
Маніпуляція намірами (prompt injection)	Відсутнє	Частково	Частково	Повністю
Агрегаційний витік	Відсутнє	Відсутнє	Частково	Повністю
Некоректна робота моделі	Відсутнє	Відсутнє	Відсутнє	Частково
Надмірні повноваження агента	Відсутнє	Частково	Частково	Повністю
Порушення контексту доступу	Відсутнє	Відсутнє	Частково	Повністю

На основі таблиці можна розрахувати узагальнений показник виявлення (де «Повністю» це 1, «Частково» = 0.5, «Відсутнє» = 0) для кожного інструменту по п'яти категоріях:

- DAST-сканер: 0 %
- WAF: 20 %
- Ручний пентест: 40 %
- Запропонована модель: 90 %

Ці дані узгоджуються з висновками ENISA [4], де зазначається, що стандартні інструменти тестування безпеки виявляють менше 25 % загроз, пов'язаних із логікою роботи AI-компонентів. Behl et al. [11] підтверджують, що ручний пентест охоплює лише частину логічних вразливостей у AI-інтегрованих системах через відсутність стандартизованих методик тестування семантичного рівня. Chandrasekaran et al. [12] вказують на обмеженість автоматизованих сканерів щодо виявлення загроз, що реалізуються через агрегацію даних.

Висновки та перспективи подальших досліджень

У статті обґрунтовано, що інтеграція інтелектуальних помічників у вебсистеми є архітектурним зламом з точки зору безпеки, а не лише черговим ускладненням. Принципова відмінність полягає у появі недетермінованого посередника між наміром користувача і виконанням системних операцій, що переміщує основну поверхню атак з рівня синтаксичної валідації на рівень семантичної інтерпретації - рівень, для якого класичні методи моделювання загроз (STRIDE, OWASP) структурно не призначені.

Запропонована класифікація п'яти категорій AI-орієнтованих загроз (маніпуляція намірами, агрегаційний витік, некоректна робота моделі, надмірні повноваження агента, порушення контексту доступу) і чотирирівнева модель аналізу загроз є конкретним інструментальним внеском, що дозволяє аналітикам безпеки систематично охоплювати вектори атак, невидимі для традиційного пентесту. Зокрема, рівень інтерпретації запиту є новим об'єктом аналізу, що немає відповідника у класичних вебсистемах.

Разом з тим необхідно чітко зазначити обмеження дослідження. Запропонована модель є концептуальною: класифікація загроз і рівнева декомпозиція обґрунтовані теоретично та через типові сценарії, але не верифіковані на реальних промислових системах. Крім того, поза межами дослідження залишаються кількісні характеристики ризиків (ймовірність реалізації, потенційний збиток) і специфіка конкретних архітектур AI-агентів (ReAct,

function calling, multi-agent systems), які можуть суттєво впливати на характер загроз.

З практичної точки зору, результати дослідження свідчать про необхідність доповнення існуючих стандартів безпечної розробки (OWASP ASVS, NIST SSDF) вимогами, специфічними для AI-інтеграцій: явне обмеження меж повноважень помічника (принцип мінімальних привілеїв на рівні агента), обов'язкова перехресна перевірка контексту доступу перед агрегацією даних з кількох сервісів, а також логування семантичних інтентів сесій помічника для подальшого аудиту.

Перспективи подальших досліджень охоплюють три основні напрями. По-перше, емпірична верифікація запропонованої класифікації на реальних системах з AI-помічниками - зокрема через red-teaming і аналіз інцидентів. По-друге, розробка формалізованої методики тестування безпеки AI-орієнтованих вебсистем, що охоплює рівень інтерпретації запитів. По-третє, дослідження специфіки загроз для різних архітектурних патернів AI-агентів (multi-agent, RAG, tool-use), оскільки кожен з них формує відмінну топологію поверхні атак. Ці напрями є основою для переходу від концептуальної до практично застосовної моделі безпеки AI-інтегрованих вебсистем.

Список літератури

[1] OWASP Foundation. OWASP Top 10: The Ten Most Critical Web Application Security Risks. URL: https://owasp.org/www-project-top-ten/

[2] OWASP Foundation. OWASP Web Security Testing Guide v4.2. URL: https://owasp.org/www-project-web-security-testing-guide/

[3] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0). URL: https://doi.org/10.6028/NIST.AI.100-1

[4] European Union Agency for Cybersecurity (ENISA). ENISA Threat Landscape for Artificial Intelligence. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

[5] MITRE Corporation. MITRE ATLAS: Adversarial Threat Landscape for Artificial Intelligence. URL: https://atlas.mitre.org

[6] OWASP Foundation. OWASP API Security Top 10. URL: https://owasp.org/www-project-api-security/

[7] ISO/IEC 23894: Artificial intelligence — Risk management. International Organization for Standardization, 2023.

[8] Shostack A. Threat Modeling: Designing for Security. Wiley, 2014. 624 с.

[9] Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 800 с.

[10] Russell S., Norvig P. Artificial Intelligence: A Modern Approach. 4th ed. Pearson, 2021. 1132 с.

[11] Behl A., Behl K., Behl D. Security implications of artificial intelligence in web applications. Journal of Cybersecurity and Privacy. 2022. Vol. 2, № 3. С. 512–528.

[12] Chandrasekaran M., Amirthalingam P., Palanisamy V. Security threats and mitigation techniques in artificial intelligence systems. Future Internet. 2021. Vol. 13, № 8. С. 201.

Yakymchuk Y., Marchenko Y., Kryvokulska O., Yakovenko O . Analysis and classification of web threats in systems with intelligent assistants

Abstract. *This article investigates the transformation of web threat models driven by the integration of intelligent assistants into modern distributed web systems. Unlike existing approaches that focus primarily on technical vulnerabilities, the study argues that an intelligent assistant constitutes a qualitatively new architectural attack surface for which classical threat modeling frameworks - including STRIDE and OWASP methodologies - are structurally inadequate. The core finding is that the principal risk shifts from the syntactic validation layer to the semantic interpretation layer: an intelligent assistant acts as a probabilistic intermediary between user intent and system operations, making its behavior context-dependent and non-deterministic in ways that evade automated scanners and standard penetration testing. Based on comparative analysis of traditional and AI-oriented architectures, five categories of novel threats are identified: intent manipulation (prompt injection), information leakage through data aggregation, model hallucination risks, agent over-permission, and context boundary violations. For each category, the article demonstrates why it remains invisible to DAST tools and conventional security controls. A four-layer threat modeling framework is proposed, explicitly incorporating the intent interpretation layer as a distinct attack surface without a counterpart in classical web systems. The proposed framework extends STRIDE by treating the intelligent assistant as an independent actor with its own decision logic. The study acknowledges its primary limitation: the classification and model are conceptually grounded and scenario-validated rather than empirically verified on production systems. Directions for future work include empirical red-teaming validation, development of AI-specific security testing methodologies, and analysis of threat topology across different agent architectures (RAG, multi-agent, tool-use).*

Keywords: *web security; threat modeling; intelligent assistants; artificial intelligence; AI-based threat modeling; web system cybersecurity; web application architecture.*

Якимчук Євгеній Анатолійович, асистент кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Yevhenii Yakymchuk, Assistant of the Department of Cybersecurity, State University «Kyiv Aviation Institute»

Марченко Ярослав Володимирович, асистент кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Yaroslav Marchenko, Assistant of the Department of Cybersecurity, State University «Kyiv Aviation Institute»

Кривокульська Ольга Олексіївна, старший викладач кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Olha Kryvokulska, senior lecturer of the Department of Cybersecurity, State University «Kyiv Aviation Institute»

Яковенко Олеся Леонідівна, старший викладач кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Olesia Yakovenko,, senior lecturer of the Department of Cybersecurity, State University «Kyiv Aviation Institute»