УДК 004.056.52

Гапон А. О., **Бабич М. Г.**

ЗАСТОСУВАННЯ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ У ПРОГРАМНОМУ КОДІ

Стрімке зростання обсягів та складності програмного забезпечення призводить до пропорційного збільшенням кількості потенційних вразливостей. Традиційні підходи до виявлення дефектів безпеки, такі як статичне (SAST) та динамічне (DAST) тестування, мають суттєві обмеження, вони часто генерують велику кількість хибнопозитивних результатів (false positives) і базуються на жорстко заданих правилах (сигнатурах), що робить їх неефективними проти вразливостей нульового дня. За таких умов інтеграція методів штучного інтелекту та машинного навчання в процеси аналізу програмного коду стає важливим етапом розвитку кібербезпеки.

Метою дослідження є обгрунтування наукової доцільності використання методів штучного інтелекту для підвищення ефективності виявлення вразливостей програмного забезпечення в умовах зростання складності сучасних інформаційних систем та обмежень традиційних інструментів аналізу безпеки.

Штучний інтелект у сфері пошуку вразливостей програмного забезпечення базується не на простому зіставленні шаблонів або перевірці регулярних виразів, а на глибокому аналізі семантики, структури та поведінки програмного коду [1]. Сучасні підходи передбачають перетворення вихідного коду у формалізовані математичні представлення, придатні для подальшої обробки алгоритмами машинного навчання. З цією метою програмний код аналізується у вигляді абстрактних синтаксичних дерев, графів потоку керування та графів залежностей даних, що дозволяє відобразити як синтаксичну будову програми, так і логічні зв’язки між її компонентами. Отримані

структурні елементи перетворюються у багатовимірні векторні представлення за допомогою спеціалізованих моделей, таких як Code2Vec або графові нейронні мережі, завдяки чому система здатна виявляти нетипові патерни та структурні аномалії, характерні для потенційних вразливостей.

Важливу роль у сучасному аналізі безпеки відіграють великі мовні моделі, спеціально адаптовані для роботи з програмним кодом, зокрема CodeBERT та StarCoder. Такі моделі навчаються на значних обсягах відкритого вихідного коду та здатні інтерпретувати код подібно до природної мови, вони враховують контекст використання змінних, логіку виконання функцій, а також приховані залежності між окремими модулями програми [2]. Завдяки цьому стає можливим виявлення складних логічних помилок, зокрема порушень механізмів контролю доступу, некоректної обробки станів або потенційно небезпечних сценаріїв взаємодії між компонентами, які часто залишаються поза увагою традиційних статичних аналізаторів.

Окремим напрямом є інтелектуалізація динамічного тестування, зокрема застосування методів AI-fuzzing. У цьому випадку штучний інтелект використовується для автоматизованої генерації вхідних даних, що подаються на виконання програмі з метою виявлення нестандартних або аварійних сценаріїв роботи. На відміну від класичного фазингу, де тестові значення формуються випадково, інтелектуальні системи використовують методи навчання з підкріпленням або генеративні моделі для цілеспрямованого формування таких наборів даних, які забезпечують максимальне покриття виконуваного коду. Це значно підвищує ймовірність виявлення критичних дефектів, зокрема переповнення буфера, помилок керування пам’яттю або некоректної обробки виняткових ситуацій.

Впровадження технологій штучного інтелекту в процеси безпечної розробки програмного забезпечення в межах концепції DevSecOps суттєво розширює можливості виявлення, аналізу та усунення вразливостей на всіх етапах життєвого циклу програмного продукту. На відміну від традиційних засобів контролю безпеки, які переважно орієнтуються на заздалегідь визначені сигнатури та правила пошуку відомих дефектів, інтелектуальні моделі здатні аналізувати програмний код з урахуванням його семантичних характеристик, логічних зв’язків і контексту виконання. Це дозволяє ідентифікувати нові типи вразливостей, які раніше не були формалізовані в базах відомих загроз, але мають структурну або функціональну подібність до вже відомих небезпечних конструкцій.

Попри значний потенціал, використання штучного інтелекту супроводжується низкою обмежень. Однією з ключових проблем залишається недостатня пояснюваність результатів глибоких моделей, оскільки більшість нейронних мереж не можуть надати формального обґрунтування причин класифікації певного фрагмента коду як небезпечного. Висока залежність від якості навчальних даних також створює ризик перенесення помилок із навчальних вибірок у практичні результати аналізу. Додатковою проблемою є обмеження контекстного вікна під час аналізу великих програмних систем, що ускладнює міжпроцедурний аналіз і може призводити до пропуску вразливостей у складних розподілених архітектурах [3].

Перспективним напрямом є розвиток гібридних систем типу Human-in-the-Loop, у яких штучний інтелект використовується як інструмент підтримки експертного аудиту, а остаточне рішення приймається фахівцем з кібербезпеки. Подальші дослідження доцільно спрямовувати на підвищення пояснюваності моделей, створення спеціалізованих датасетів для навчання, а також інтеграцію ШІ в безперервні конвеєри безпечної розробки програмного забезпечення.

Штучний інтелект формує нову парадигму забезпечення безпеки програмного забезпечення, переходячи від реактивного сигнатурного контролю до проактивного семантичного аналізу. Незважаючи на існуючі технічні обмеження, сучасні інтелектуальні методи вже демонструють високу ефективність у масштабуванні

перевірок, автоматизації рутинних процесів та виявленні складних вразливостей, що робить їх важливим компонентом майбутніх систем кіберзахисту.

Список використаних джерел

1. Бабич М. Г. Розпізнавання програмного коду, створеного штучним інтелектом / М. Г. Бабич, Д. О. Цемма // Проблеми інформатизації: тези доп. тринадцятої міжнар. наук.-техн. конф., 27-28 листопада 2025 р., м. Баку, м. Харків, м. Бельсько-Бяла: [у 4 т.]. Т. 3: секції 4 / Ін-т систем управління МНО Азербайджанської республіки, Національний технічний університет "Харківський політехнічний інститут", Харківський національний університет радіоелектроніки [та ін.]; орг. ком.: співголови: Гашимов Ельшан Гіяс огли [та ін.] – Харків: НТУ "ХПІ", 2025. – С. 46.

2. Колощук М.С., Дячук О.Ю., Окунькова О.О., Пірог О.В. (2024). "Інструменти штучного інтелекту для автоматизації тестування на проникнення." Технічна інженерія, № 2(94), с. 121-128.

3. Pan, C., Lu, M., & Xu, B. (2021). An empirical study on software defect prediction using codebert model. *Applied Sciences*, *11*(11), 4793.