КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (33), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.33.1232
УДК 004.056

Притула Андрій Вікторович
аспірант кафедри захисту інформації
Вінницький національний технічний університет, Вінниця, Україна
ORCID: 0009-0006-9632-0712
andrik.pritula@gmail.com

Куперштейн Леонід Михайлович
к.т.н, доцент кафедри захисту інформації
Вінницький національний технічний університет, Вінниця, Україна
ORCID: 0000-0001-6737-7134
kupershtein.lm@gmail.com

МОДЕЛЮВАННЯ СЦЕНАРІЇВ КІБЕРАТАК ЯК МАРКОВСЬКОГО ПРОЦЕСУ ІЗ СЕМАНТИЧНО ОБМЕЖЕНИМ ПРОСТОРОМ ДІЙ

Анотація. Запропоновано формальну модель подання сценаріїв кібератак у вигляді марковського процесу прийняття рішень, у якій, на відміну від статичних графів атак, явно задається динаміка зміни станів системи залежно від виконаних кроків атаки, а множина допустимих дій формується з урахуванням семантичних залежностей між кроками, зокрема залежностей типу І (AND) та АБО (OR). Запропонований підхід забезпечує часову інтерпретацію сценаріїв через метрику time-to-compromise (TTC) та дозволяє описувати як прості, так і складні багатокрокові траєкторії компрометації. Модель поєднує динамічне MDP-подання з інваріантним графовим представленням станів, що будується з використанням механізмів графових нейронних мереж. Експериментальне дослідження проведено на множині стохастично-згенерованих MAL-графів, узгоджених із відкритими моделями атак та веб-датасетами, і включає порівняння з базовими графовими методами та методами навчання з підкріпленням без семантичних обмежень. Отримані результати показують, що запропонований підхід забезпечує суттєве скорочення середнього часу до компрометації та зменшує дисперсію результатів, що свідчить про підвищення стабільності навчання. Доведено, що введення семантично обмеженої множини допустимих дій повністю усуває нерелевантні переходи та суттєво підвищує частку успішних сценаріїв компрометації. Найбільший виграш спостерігається на глибоких багатокрокових траєкторіях атак із домінуванням AND-залежностей, де семантична структура графа має визначальний вплив на простір доступних рішень. Практичне значення полягає в можливості застосування моделі для кількісного оцінювання сценаріїв кібератак, ранжування траєкторій компрометації та підтримки прийняття рішень, а також інтеграції у системи автоматизованого тестування на проникнення та навчальні кіберполігони.

Ключові слова: марковський процес; прийняття рішень; кібербезпека; загроза; атака; навчання з підкріпленням; нейронна мережа; моделювання; машинне навчання; штучний інтелект.

ВСТУП

Сучасні сценарії кібератак дедалі частіше мають багатокроковий характер і реалізуються як послідовності взаємозалежних дій, для яких важливими є не лише досяжність цілі, а й порядок виконання кроків, їх часові характеристики та зміна простору доступних дій у ході компрометації. Водночас аналіз графів атак лишається одним із базових інструментів формального опису атак, проте сучасні огляди прямо підкреслюють його проблеми зі складністю, циклічністю, масштабованістю та практичним використанням у великих системах [1]. Аналогічно, у роботі [2] відзначено, що графи атак є корисними для оцінювання вразливостей, але водночас надто великими й складними для ефективного аналізу, тому потребують інтелектуалізованих методів опрацювання. У роботах [3-5], орієнтованих на реконструкцію векторів атак засобами GNN-LSTM, також наголошується, що сучасні кібератаки є динамічними й багатокроковими, а традиційні реактивні підходи не забезпечують достатнього рівня структурно-часового моделювання.

Постановка проблеми. Звідси випливає актуальна наукова задача, яка в літературі досі не має цілісного розв'язання. Існуючі графові моделі добре подають структуру потенційних атак, але переважно залишаються статичними. Підходи на основі навчання з підкріпленням моделюють динаміку вибору дій, але часто не мають формального механізму семантичного обмеження простору дій відповідно до логіки кроків атаки [6-8]. Графові нейронні мережі ефективно будують представлення графових структур, але зазвичай використовуються для класифікації або виявлення загроз, а не для формального планування багатокрокових траєкторій компрометації [9-11]. Внаслідок цього залишається невирішеною задача побудови єдиної формальної моделі, яка б одночасно враховувала динаміку зміни станів системи, семантичні залежності між кроками атаки та часову оцінку траєкторій через час до компрометації. Розв'язання цієї проблеми має безпосереднє значення для задач автоматизованого тестування безпеки, кіберполігонів та інструментів моделювання атак, де потрібні не лише структурні, а й динамічно-часові оцінки сценаріїв компрометації.

Аналіз останніх досліджень і публікацій. У сучасній літературі можна виокремити три основні напрями, які частково розв'язують зазначену задачу. Перший напрям пов'язаний із використанням графів атак як базового формалізму для подання структури можливих сценаріїв компрометації [1, 12-14]. Другий напрям охоплює застосування методів навчання з підкріпленням для моделювання динаміки вибору атакувальних дій і побудови оптимальних стратегій [2, 6-8]. Третій напрям представлений підходами, що використовують графові нейронні мережі для побудови ефективних представлень атакувальних структур і виявлення прихованих залежностей [5, 9-11]. Незважаючи на суттєві досягнення в кожному з цих напрямів, їх результати залишаються фрагментованими, що зумовлює доцільність їх окремого критичного аналізу.

Перший релевантний напрям базується на дослідженнях, у яких графи атак застосовуються як основний формалізм опису атакувальних шляхів. В статті [1] графи атак систематизовано як модельно-орієнтований інструмент аналізу мережевої безпеки, що дозволяє працювати з орієнтованим графовим поданням вразливостей і залежностей між експлойтами. В роботі [12] графи атак використовуються для багатоступеневої оцінки ризику і кількісного ранжування безпеки мережі. В статті [13] запропоновано алгоритм пошуку шляху атаки з мінімальним часом у циклічних AND/OR-графах атак. В роботі [14] графи атак адаптовано до мікросервісної архітектури, що демонструє розширення класичного формалізму на сучасні розподілені системи. Ці роботи переконливо показують цінність графів атак як структурної моделі, проте водночас виявляють їх спільне обмеження. В центрі уваги лишається або генерація графа, або пошук шляху в уже побудованій структурі, або кількісна оцінка ризику на статичному графі. Навіть коли враховуються AND/OR-вузли та часові ваги, траєкторія атаки переважно інтерпретується як шлях у графі, а не як процес послідовного прийняття рішень у просторі станів із динамічною множиною допустимих дій. Саме тому цей клас підходів не забезпечує повного переходу від статичного аналізу графів атак до динамічного моделювання сценарію компрометації.

Другий напрям пов'язаний із використанням навчання з підкріпленням для аналізу, генерації або оптимізації атакувальних стратегій. У роботі [6] граф атак аналізується через Q-learning, що фактично переносить задачу оцінювання графів атак у середовище навчання з підкріпленням. У статті [2] Q-learning застосовується для аналізу графів атак у медичному кіберфізичному середовищі з орієнтацією на пошук оптимального шляху атаки. У роботі [7] RL використовується для автоматичної генерації атакувальних шляхів і доповнюється стратегією відсікання для прискорення пошуку. В статті [8] RL інтегрується з CVSS-орієнтованою оцінкою атаки в середовищі Cyber Battle Simulation для побудови ефективних стратегій зловмисника. В роботі [15] авторами проведено систематизований аналіз підходів до тестування на проникнення з використанням навчання з підкріпленням, що дозволило виокремити переваги й обмеження існуючих методів і слугувало відправною точкою для формулювання задачі динамічного моделювання сценаріїв атак, що досліджується в цій роботі. Ці роботи демонструють важливий зсув від статичних графів до динамічного пошуку політики, однак у більшості випадків простір дій або формується евристично, або задається симулятором, або визначається через внутрішню логіку конкретного середовища без явного формального зв'язку з семантикою кроків атаки. Внаслідок цього RL-постановка або лишається тісно прив'язаною до спеціалізованого симулятора, або не забезпечує семантично обмеженого простору дій, у якому нерелевантні (переходи, що порушують семантичні передумови) кроки усувалися б на рівні моделі. Саме цей недолік є критичним, оскільки потрібно не просто навчити агента знаходити «вигідні» дії, а гарантувати, що ці дії є допустимими в логіці графу атак.

Третій напрям становлять підходи, у яких графові нейронні мережі використовуються для побудови представлень атак, інференції знань про загрози або прогнозування векторів атак. В статті [9] графові згорткові мережі (Graph Convolutional Networks, GCN) і метод агрегування сусідніх вершин GraphSAGE (Graph Sample and Aggregate) застосовано для розпізнавання мережевих атак у хмарному

середовищі на основі структурних властивостей графа. В роботі [10] графові згорткові моделі інтегруються з вбудовуванням графів знань для інференції прихованих зв'язків між базами вразливостей типу CVE, CWE і CAPEC у задачах інференції знань про атаки. У статті [11] GNN поєднуються з доменним знанням для видобування сутностей і відношень у кіберрозвідці загроз. У роботі [5] поєднання нейронних мереж типу GNN і LSTM використовується для реконструкції векторів атак на основі MITRE ATT&CK та часових послідовностей подій. Ці дослідження показали, що навчання представлень графів є перспективним інструментом для роботи зі складними кібератакувальними структурами. Основний акцент робиться на виявлення, інференцію, видобування сутностей або реконструкцію векторів атак, а не на формальному плануванні сценарію компрометації як послідовності станів і дій. Іншими словами, GNN у цих роботах слугують або класифікаційним, або прогностичним інструментом, але не є частиною MDP-подання сценарію атаки із семантично обмеженим простором дій і часовою метрикою TTC. Саме тому ці підходи можна розглядати як важливе джерело методів представлення, але не як повне розв'язання задачі моделювання атакувальних траєкторій.

Таким чином, аналіз джерел показав, що статичні підходи на основі графів атак добре моделюють структуру можливих шляхів, підходи на основі RL вводять динаміку вибору дій, а підходи на основі GNN забезпечують виразне навчання представлень графів, проте жоден із цих напрямів окремо не розв'язує задачу формального подання сценаріїв кібератак як процесу зміни станів із семантично обмеженим простором дій і часовою інтерпретацією траєкторій через TTC. Саме це і визначає постановку задачі цієї статті, у якій пропонується MDP-модель сценаріїв кібератак, що поєднує динаміку переходів, семантичні залежності між кроками атаки і кількісне оцінювання траєкторій.

Мета статті. Метою є підвищення ефективності моделювання сценаріїв кібератак шляхом зменшення часу до компрометації та побудови семантично коректних траєкторій атак у межах єдиної формальної моделі. Для досягнення поставленої мети в роботі передбачено формалізацію сценаріїв кібератак у вигляді марковського процесу прийняття рішень, визначення множини допустимих дій з урахуванням семантичних залежностей між кроками атаки, введення часової інтерпретації сценаріїв через метрику TTC, розроблення інваріантного графового представлення станів, побудову експериментального середовища на основі MAL-графів і відкритих джерел даних, а також експериментальну оцінку ефективності запропонованої моделі у порівнянні з базовими методами.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

У сучасних дослідженнях кібербезпеки особливого значення набуває узгодження алгоритмів аналізу графових структур і методів навчання з підкріпленням у межах єдиної формальної постановки. Класичні графи атак забезпечують наочне подання можливих шляхів компрометації інформаційно-комунікаційної системи, однак у загальному випадку залишаються статичними моделями, у яких аналіз досяжності цільових станів не супроводжується явним описом динаміки вибору наступного кроку атаки. Така постановка є обмеженою в задачах автоматизованого тестування безпеки веб-застосунків, де принциповими є не лише факт існування траєкторії компрометації, а й порядок виконання кроків, часові характеристики проходження графа, залежність множини допустимих дій від поточного стану системи та можливість побудови адаптивної стратегії дослідження. Саме тому доцільним є перехід від статичного подання до динамічного середовища, у якому сценарій компрометації розглядається як послідовність станів і дій, а вибір наступного кроку формулюється як задача оптимізації. Для формалізації графів атак у даному дослідженні застосовується метамова Meta Attack Language (MAL) [16], яка забезпечує уніфікований формат представлення властивостей інформаційних систем у контексті їх кіберзахищеності та задає чіткі правила побудови графів атак. Відповідно, перехід від статичного MAL-графа атак до його інтерпретації як динамічного середовища становить ключовий крок формалізації, що лежить в основі цього дослідження.

Нехай задано MAL-граф атаки $G = (V, E)$ де V – множина вершин, що відповідають крокам атаки, а $E \subseteq V \times V$ – множина орієнтованих ребер, які задають відношення досяжності між ними. Кожна вершина $v \in V$ має семантичну структуру вигляду $v = (\text{class}(v), \text{inst}(v), \text{step}(v))$, де $\text{class}(v)$ – клас активу MAL-моделі (наприклад, $Application$, $Network$), до якого належить відповідний компонент системи, $\text{inst}(v)$ – конкретний екземпляр цього класу в досліджуваній системі (наприклад, окремий сервіс, вузол або хост), $\text{step}(v)$ – елементарна дія (крок атаки), визначена у межах відповідного класу MAL-специфікації та доступна для виконання над цим екземпляром. Така структура дозволяє безпосередньо інтерпретувати вершину як конкретну дію атакуючого агента над певним компонентом системи. Наприклад, вершина $Application.Service2.networkRequestConnect$ відповідає встановленню мережевого з'єднання з другим екземпляром сервісу, тоді як вершина $Network.Network2.successfulAccess$ інтерпретується як успішне отримання доступу до відповідного

мережевого вузла. Таким чином, граф G є не просто топологічною структурою, а формалізованим описом процесу експлуатації вразливостей у веб-орієнтованій інформаційно-комунікаційній системі (ІКС), де вершини кодують змістовні кроки атаки, а ребра відображають семантично допустимі залежності між ними. Саме ця обставина робить MAL-граф придатним не лише для статичного аналізу, а й для побудови середовища послідовного прийняття рішень.

Ключовою проблемою використання таких графів у задачах навчання з підкріпленням є залежність числового подання вершин від їхніх довільних ідентифікаторів. Якщо два підграфи є ізоморфними й відрізняються лише порядком нумерації вузлів, їхнє пряме числове подання буде різним, хоча семантика відповідних фрагментів атаки залишається незмінною. Це створює небажану залежність моделі від синтаксичних артефактів подання графа та погіршує її здатність до узагальнення між різними інстанціями MAL-графів.

Для усунення цієї залежності кожна вершина MAL-графа переводиться у числовий вектор ознак, який описує її не за індексом, а за локальним контекстом: типом компонента MAL-моделі, рівнем привілеїв, типом вразливості, атрибутами сусідніх вершин і часовими характеристиками відповідного кроку атаки. Для побудови таких векторів у роботі використовується стандартний механізм Graph Attention Network (GAT) [17], який для кожної вершини зважено агрегує ознаки її безпосередніх сусідів у MAL-графі, причому ваги агрегації обчислюються на основі семантичної близькості вершин, а не їхнього порядкового номера. GAT за побудовою використовує кілька паралельних голів уваги, що дозволяє одночасно враховувати різні типи локальних залежностей мережеві зв'язки, привілейовані переходи та структуру доступу до даних. Оскільки агрегація по сусідах є множинною, а не позиційною, результат не залежить від перенумерації вершин: для двох ізоморфних фрагментів атаки на кшталт $NetworkAccess \rightarrow Login \rightarrow PrivilegeEscalation$ отримуємо узгоджені векторні представлення незалежно від їх індексації. Саме ця інваріантність дозволяє моделі узагальнюватись на нові MAL-графи за структурно-семантичними, а не синтаксичними ознаками. У контексті MAL-графів це означає, що зміст кожної вершини визначається не лише її власним типом, а й тим, які попередні кроки пов'язані з нею у графі. Наприклад, крок PrivilegeEscalation інтерпретується з урахуванням того, чи вже реалізовані отримання доступу, автентифікація або експлуатація вразливості, які передують йому в конкретному сценарії. У результаті для кожної вершини формується латентне векторне представлення, яке відображає її роль у сценарії атаки разом із локальним контекстом.

Побудовані у такий спосіб векторні представлення вершин, сформовані на основі їхніх семантичних атрибутів $(class(v), inst(v), step(v))$ та локального оточення $N(v)$, утворюють вхід марковської моделі. Вони визначають опис стану s_t і дозволяють ідентифікувати, які кроки атаки вже реалізовані, які залишаються допустимими в поточному стані та які переходи можуть бути виконані далі. Таким чином, GAT у запропонованій моделі виконує роль механізму, що поєднує семантичну структуру MAL-графа з поданням станів і дій у марковському процесі прийняття рішень (рис. 1).

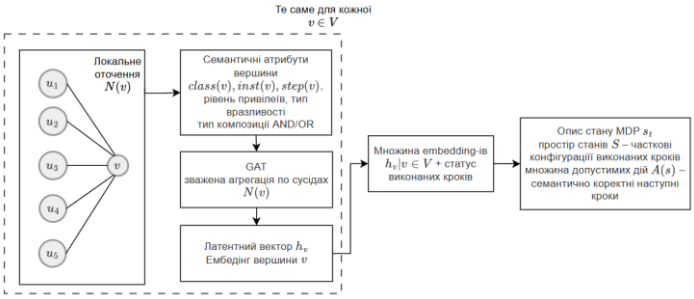

Рис. 1. Схема процесу формування векторного представлення вершини MAL-графа на основі семантичних атрибутів та її локального графового оточення

Побудоване графове представлення використовується далі для формалізації сценарію атаки як марковського процесу прийняття рішень:

$$\mathcal{M} = (S, A, P, R, \gamma) \tag{1}$$

де S – простір станів, A – простір дій, P – перехідний розподіл, R – функція винагороди, $\gamma \in [0,1)$ – коефіцієнт дисконтування. На відміну від статичного графового аналізу, така постановка дозволяє розглядати процес компрометації як послідовність виборів агента в динамічному середовищі. Стан системи задається булевим відображенням $s: V \rightarrow \{0,1\}$, де $s(v) = 1$, якщо крок атаки v уже реалізовано, і $s(v) = 0$ – інакше. Еквівалентно, стан можна трактувати як підмножину $s \subseteq V$ уже виконаних кроків

атаки. Наприклад, стан $s = NetworkAccess, Login$ означає, що атакуючий уже отримав доступ до мережі та пройшов автентифікацію, але ще не виконав подальші дії, такі як отримання можливості виконання коду або ескалація привілеїв. У такий спосіб кожен стан кодує поточний рівень компрометації системи, а весь простір S охоплює всі можливі конфігурації частково реалізованих сценаріїв атаки. Це подання узгоджується з логікою MAL, де виконані кроки атаки змінюють досяжність наступних кроків і тим самим змінюють саму конфігурацію середовища.

Дія в цій постановці відповідає вибору нового кроку атаки $a \in V$, який потенційно може бути виконаний у поточному стані. Однак, на відміну від класичних MDP, множина допустимих дій не є сталою і визначається семантикою графа. Для кожної вершини $v \in V$ вводиться множина її передумов $pa(v) = u \in V \mid (u, v) \in E$, яка задає кроки атаки, що повинні бути реалізовані раніше. У випадку AND-залежності (кон'юнктивної залежності) вершина v може бути активована лише за умови, що всі її батьківські вузли $pa(v)$ вже досягнуті (тобто всі необхідні передумови виконані). У випадку OR-залежності (диз'юнктивної залежності) достатньо доягнення хоча б одного з батьківських вузлів.. Відповідно, множина допустимих дій у стані s задається співвідношенням

$$A(s) = \{v \in V \setminus s | (\tau(v) = AND \Rightarrow pa(v) \subseteq s) \wedge (\tau(v) = OR \Rightarrow pa(v) \cap s \neq \emptyset)\} \quad (2)$$

де $\tau(v)$ визначає тип композиції передумов для вузла v. Саме (2) забезпечує явне врахування семантичних залежностей між кроками атаки та усуває нерелевантні переходи. Наприклад, якщо крок $PrivilegeEscalation$ залежить від одночасного досягнення $Login$ та $CodeExecution$, то за стану, у якому виконано лише автентифікацію, відповідна дія не входить до $A(s)$. Лише після реалізації обох передумов цей крок стає допустимим. Отже, множина дій є функцією поточного стану системи, а динаміка атаки визначається не абстрактним переходом по ребрах графа, а семантично коректним вибором наступного кроку. Саме це становить одну з принципових відмінностей запропонованої моделі від традиційних статичних графи атак, де наявність ребра часто ототожнюється з безумовною досяжністю наступного вузла.

Перехід між станами визначається активацією вибраної допустимої дії. У базовій конструктивній постановці, що відповідає логіці досяжності кроків атаки, перехід може бути поданий як

$$s_{t+1} = s_t \cup a_t, \quad (3)$$

де $a_t \in A(s_t)$. Співвідношення (3) відображає монотонний рекурентний характер процесу компрометації. Якщо крок атаки вже реалізовано, він залишається активованим на всіх наступних етапах вектору атаки. Із цього безпосередньо випливає властивість $s_t \subseteq s_{t+1}$ для будь-якого t, тобто процес атаки в межах даної моделі є незворотним у просторі досягнутих кроків атаки. Така конструкція є доречною саме для формалізації логіки проходження MAL-графа, де ключовим є не стохастичне скасування досягнутих станів, а накопичення умов, які відкривають доступ до нових кроків атаки. Отже, будь-яка траєкторія $\tau = (s_0, a_0, s_1, a_1, \dots, s_T)$ однозначно відповідає послідовності коректних з погляду MAL кроків атаки. Інакше кажучи, дія, яка порушувала б семантичні передумови, просто не могла б бути згенерована механізмом (2), а тому нерелевантні сценарії усуваються на рівні самої моделі, а не на етапі постфактум-фільтрації. Це є принципово важливим для задач автоматизованого тестування безпеки, оскільки модель повинна не лише знаходити траєкторії, а й гарантувати їх семантичну валідність.

Для кількісної інтерпретації траєкторій вводиться часова характеристика часу до компрометації. Нехай кожному кроку атаки $v \in V$ відповідає випадкова величина $T_v = \mathrm{TTC}(v)$, що характеризує очікуваний час, необхідний для реалізації цього кроку. У межах поточної постановки використовується саме математичне сподівання цієї величини, тобто модель працює з часовими оцінками на рівні кроків, а не з конкретними реалізаціями випадкових затримок. Тоді для довільної траєкторії $\tau = (v_1, \dots, v_k)$ сумарний час компрометації визначається як $T(\tau) = \sum_{i=1}^{k} T_{v_i}$.

У межах марковської постановки ця величина інтерпретується через функцію винагороди. Зокрема, для кроку $a \in A(s)$ можна задати

$$R(s, a) = -\mathbb{E}[T_a] \quad (4)$$

що означає штрафування тривалих кроків атаки та, відповідно, орієнтацію на мінімізацію очікуваного часу компрометації. Така інтерпретація є природною для задач автоматизованого тестування безпеки. Швидші сценарії, які приводять до досягнення критичних станів за менший час, становлять більший практичний інтерес для аналізу системи, що атакується. Водночас важливо підкреслити, що у статті TTC використовується як базова часова метрика сценарію, а не як окрема розширена схема формування

функції винагороди. Завдяки цьому зберігається концептуальна незалежність від подальшої спеціалізованої побудови функції винагороди.

Після введення станів, дій, переходів та винагороди задача побудови оптимальної стратегії атаки формулюється як задача навчання з підкріпленням. Політика агента $\pi: S \to A$ ставить кожному стану $s \in S$ у відповідність дію $a \in A(s)$, тобто визначає вибір наступного кроку атаки з урахуванням семантичних обмежень графа. Метою є максимізація очікуваної сумарної дисконтованої винагороди

$$J(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)] \to max. \quad (5)$$

З урахуванням (4) максимізація (5) еквівалентна мінімізації очікуваного часу компрометації системи. Для аналізу цієї задачі вводяться функція цінності стану $V^{\pi}(s)$ та функція дії-цінності $Q^{\pi}(s,a)$, які визначають відповідно очікувану сумарну винагороду для даного стану та для пари «стан–дія». У термінах кібербезпеки $Q^{\pi}(s,a)$ інтерпретується як оцінка доцільності вибору кроку атаки a у стані s з погляду подальшого часу досягнення цільових компрометованих конфігурацій. Саме така інтерпретація переводить абстрактну RL-постановку в площину аналізу атакувальних стратегій. Значення Q-функції показує не просто «корисність» дії, а її очікуваний внесок у скорочення часу до цілі за умови, що решта траєкторії також вибудовується оптимально.

Оптимальна політика визначається як така, що в кожному стані вибирає дію з найбільшим значенням оптимальної Q-функції:

$$\pi^*(s) = arg \max_{a \in A(s)} Q^*(s,a) \quad (6)$$

Принципово важливо, що максимум у (6) береться не по всьому абстрактному простору дій, а саме по множині $A(s)$, індукованій семантикою MAL-графа. Це означає, що оптимізація відбувається лише серед семантично допустимих кроків атаки, а отже, алгоритм не може обирати дії, які суперечать логіці побудови сценарію компрометації. Для оцінювання $Q^*(s,a)$ використовується алгоритм Q-навчання, адаптований до станозалежної множини допустимих дій. Його ітеративне оновлення має вигляд:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t \left[r_t + \gamma \max_{b \in A(s_{t+1})} Q_t(s_{t+1}, b) - Q_t(s_t, a_t) \right], \quad (7)$$

де $\alpha_t \in (0,1]$ – коефіцієнт навчання, а $r_t = R(s_t, a_t)$ – миттєва винагорода, отримана агентом у стані s_t після виконання дії a_t, яка в цій постановці інтерпретується як зміна оцінки часу до компрометації TTC. На відміну від класичного запису Q-навчання, у (7) максимум знову береться за множиною допустимих дій $A(s_{t+1})$, а не за всім простором A. Саме ця модифікація забезпечує узгодження алгоритмічного рівня з формальною моделлю сценаріїв атаки. На практиці це означає, що оцінка майбутньої вигоди від переходу в новий стан не враховує нереалістичні кроки, які не можуть бути виконані через невиконання передумов. За стандартних умов збіжності та за умови достатнього дослідження простору станів і дій співвідношення (7) приводить до оптимальної Q-функції.

Однак у випадку реальних веб-орієнтованих ІКС простір станів S має експоненційну розмірність, оскільки кожен стан відповідає деякій підмножині реалізованих кроків атаки. Це робить табличне подання $Q(s,a)$ практично непридатним. Для подолання цього обмеження використовується нейромережева параметризована апроксимація функції дії-цінності $Q(s,a;\theta)$, де θ – параметри нейронної мережі. Принциповим є те, що вхід цієї мережі формується не з сирих індексів вузлів, а з інваріантного графового представлення, побудованого за допомогою GAT. Отже, апроксимація $Q(s,a;\theta)$ поєднує два рівні узагальнення. По-перше, інваріантність до перенумерації вузлів, по-друге, перенесення знань на структурно подібні, але не тотожні екземпляри MAL-графів. Саме тут графове подання й RL-постановка остаточно зливаються в єдиний алгоритмічний аппарат. Без інваріантного представлення мережа була б чутливою до довільних перестановок вузлів, а без марковської формалізації саме графове представлення не мало б чіткої оптимізаційної інтерпретації.

Одним із найбільш придатних алгоритмів у такій постановці є Deep Q-learning, який поєднує ідею Q-навчання із глибинною нейронною апроксимацією та стабілізаційними механізмами, зокрема буфером досвіду і цільовою мережею. У цьому випадку параметри θ оновлюються шляхом мінімізації квадратичної функції втрат

$$(\theta) = \mathbb{E}_{(s,a,r,s')}[(r + \gamma \max_{b \in A(s')} Q(s', b; \theta^-) - Q(s,a;\theta))^2] \quad (8)$$

де θ^- – параметри цільової мережі, а s' – наступний стан системи, що досягається після виконання дії a у стані s. Співвідношення (8) зберігає ту саму принципову властивість, що й (7): оцінювання

цільового значення здійснюється лише за множиною семантично допустимих дій. Це означає, що і в глибинній постановці логіка MAL-графа зберігається не як зовнішнє обмеження, а як внутрішня частина алгоритму. Вибір дії в процесі навчання здійснюється за ε – жадібною політикою. Це означає, що з імовірністю $1-\varepsilon$ агент обирає дію, що максимізує $Q(s,a;\theta)$ на множині $A(s)$, а з імовірністю ε – випадкову дію з тієї самої множини. Такий механізм забезпечує баланс між дослідженням нових траєкторій атаки та використанням уже вивченої інформації про найефективніші шляхи компрометації. Алгоритмічно це реалізується як ітеративна взаємодія агента з MAL-середовищем. Тобто на кожному кроці формується поточний стан s_t, із множини $A(s_t)$ обирається допустима дія a_t, виконується перехід до s_{t+1} відповідно до (3), обчислюється винагорода r_t згідно з (4), після чого параметри моделі оновлюються на основі вибірки з буфера досвіду за правилом (8). Періодичне копіювання параметрів основної мережі в цільову стабілізує процес навчання, а використання буфера досвіду зменшує кореляцію між послідовними переходами. Така процедура і становить алгоритмічну основу практичної реалізації запропонованої моделі у вигляді адаптованої Deep Q-Network (DQN) для аналізу сценаріїв атак у веб-застосунках.

У підсумку запропонований підхід утворює єдину формальну конструкцію, у якій MAL-граф атаки подається як марковське середовище з інваріантним графовим представленням, станозалежною множиною допустимих дій та часовою інтерпретацією траєкторій через TTC. На відміну від статичних графів атак, у цій моделі явно задається динаміка зміни станів системи залежно від виконаних кроків атаки, а простір дій обмежується семантичними залежностями між кроками, зокрема залежностями типу AND/OR. Це дозволяє описувати сценарії компрометації як послідовності станів із часовою інтерпретацією та кількісно характеризувати їх за метрикою TTC, виокремлюючи як прості, так і складні багатокрокові траєкторії атак у межах єдиного математичного апарату.

МЕТОДИКА ДОСЛІДЖЕННЯ

Експериментальне дослідження спрямоване на перевірку того, що формалізація сценаріїв кібератак у вигляді марковського процесу прийняття рішень не лише коректно відтворює логіку проходження MAL-графа, а й забезпечує вимірюваний виграш у часових характеристиках компрометації, коректності траєкторій та стабільності навчання. Критично важливим є те, що верифікації підлягає не загальна теза про корисність навчання з підкріпленням, а конкретна конструкція, у якій інваріантне графове представлення, станозалежна множина допустимих дій $A(s)$ та метрика TTC працюють як взаємопов'язані компоненти одного формального апарату.

Для експериментальної верифікації запропонованого підходу використано п'ять методів, які охоплюють як повну конфігурацію моделі, так і базові підходи з різним рівнем врахування семантики MAL-графа та часової оптимізації, що дозволяє ізолювати внески окремих компонентів і зіставити запропоновану конструкцію з характерними альтернативами.

1. Proposed_MDP_As_GAT (далі – Proposed) – запропонований у цій статті метод, який реалізує повну конфігурацію моделі. Сценарій кібератаки подається як марковський процес прийняття рішень із простором станів S, станозалежною множиною допустимих дій $A(s)$, індукованою AND/OR-залежностями MAL-графа, та функцією винагороди, заснованою на метриці time-to-compromise. Стан середовища кодується інваріантним графовим представленням, побудованим механізмом GAT, що усуває залежність від довільних ідентифікаторів вузлів. Політика агента формується за допомогою адаптованого алгоритму Deep Q-learning, у якому максимум цільового значення в (7) та (8) береться виключно за множиною $A(s)$, тобто семантично допустимі кроки враховуються не як постфільтр, а як структурна властивість моделі.
2. MDP_As_no_GAT – спрощений варіант запропонованого методу, у якому зберігається вся MDP-постановка зі станозалежною множиною $A(s)$, функцією винагороди за TTC та алгоритмом Q-learning, але інваріантне графове представлення GAT замінено на пряме ознакове кодування вузлів без агрегації за околом. Цей метод дозволяє ізолювати внесок саме графового представлення: будь-яка різниця між Proposed і MDP_As_no_GAT характеризує вплив GAT, тоді як спільні риси обох методів відображають ефект від самого семантичного обмеження простору дій.
3. RL_no_constraints (далі – RL без обмежень) – базовий RL-метод без семантичних обмежень. Побудований на тій самій архітектурі Deep Q-learning і тій самій функції винагороди за TTC, що й Proposed, але множина дій у кожному стані відповідає всьому простору вершин графа без урахування AND/OR-передумов. Вибір такого базового методу є принциповим: він дозволяє перевірити, чи справді виграш Proposed породжується семантичною структурою $A(s)$, а не просто застосуванням навчання з підкріпленням до задачі.

4. Shortest_path_static_graph (далі – shortest path) – графовий базовий метод, що відповідає класичному аналізу графів атак. Траєкторія компрометації обчислюється як найкоротший шлях у MAL-графі з урахуванням ваг ребер, пропорційних очікуваним значенням TTC кроків атаки.

5. Random_policy (далі – випадкова стратегія) – контрольний метод, у якому вибір дії в кожному стані здійснюється рівномірно випадково з повного простору вершин без урахування семантичних обмежень. Цей метод не претендує на практичну ефективність і слугує нижньою межею порівняння: його результати показують, який рівень часових і структурних показників досягається без жодної стратегічної логіки, і дозволяють коректно інтерпретувати абсолютні значення метрик для решти методів.

Усі результати одержано на множині з десяти сценаріїв, сформованих не довільно, а шляхом генерування MAL-графів на основі відкритих моделей атак і відтворюваного процесу генерації. Кількість сценаріїв обмежено десятьма як компроміс між різноманітністю структур і обчислювальною складністю експериментів. Така кількість дозволяє охопити різні конфігурації графів (за розміром, глибиною та типами залежностей) і водночас забезпечити статистично надійні результати за рахунок багаторазових запусків кожного методу.

Формальна семантика кроків атаки та їх залежностей була взята з публічного репозиторію enterpriseLang [18], який реалізує доменно-специфічну мову MAL і містить типові шаблони компрометації, зокрема використання вразливостей публічно доступного застосунку, ескалацію привілеїв та латеральне переміщення. Для прив'язки цих шаблонів до веб-орієнтованого середовища прикладні вузли та їх послідовності були зіставлені з відкритими навчальними сценаріями OWASP WebGoat [19], що охоплюють SQL-ін'єкції, обхід автентифікації, небезпечне пряме посилання на об'єкт та інші характерні кроки атак веб-застосунків. Часові характеристики окремих кроків атаки і їх варіативність задавалися не вручну, а через узгодження з емпіричними розподілами, отриманими з відкритих HTTP-датасетів CSIC 2010 HTTP Dataset [20] та Malicious URL Detection Dataset [21], що дозволило відобразити реалістичну частоту, складність і тривалість типових атак. Важливо підкреслити, що зазначені відкриті джерела використовувалися не безпосередньо, а як основа для визначення статистичних розподілів кількості вузлів, ребер, глибини графа та часток AND/OR-залежностей. На основі цих розподілів здійснювалася стохастична генерація MAL-графів, які відтворюють типові структурні патерни реальних атак, але не прив'язані до конкретної інфраструктури. Для кожного сценарію змінювалися кількість екземплярів вузлів, глибина графа (в межах 5-8 рівнів), частка AND/OR-залежностей, а також конфігурація міжвузлових зв'язків. При цьому множина вершин графа складається з трьох типів вузлів: AND- та OR-вузлів, що описують кон'юнктивні й диз'юнктивні композиції передумов, і базових (листкових) вузлів без передумов, які слугують стартовими точками компрометації (наприклад, мережевий доступ або експлуатація публічно доступного застосунку) . Отримані графи трансформувалися у MDP-подання шляхом визначення множини станів S, допустимих дій $A(s)$, функції переходів і винагороди, пов'язаної з метрикою TTC.

Параметри сценаріїв (кількість вузлів і ребер, глибина, співвідношення AND/OR-залежностей) обиралися на основі аналізу типових структур атак, представлених у відкритих моделях (зокрема MITRE ATT&CK та MAL). Діапазони значень відповідають типовому розміру підграфа компрометації для окремого веб-застосунку чи мікросервіса – від компактних сценаріїв з однією лінією атаки до розгалужених багатокрокових траєкторій з альтернативними шляхами. Інтервали значень сформовано таким чином, щоб відобразити як сценарії з високою варіативністю (домінування OR-залежностей), так і сценарії з жорсткою послідовністю дій (домінування AND-залежностей). Це дозволяє оцінити поведінку методів у різних умовах складності та перевірити їх здатність до узагальнення.

Відповідно, кожен згенерований MAL-граф має $|V| \in [27,48]$ (кількість вершин графа), $|E| \in [43,88]$ (кількість ребер графа), глибину 5-8 (довжина найдовшого шляху атаки), кількість AND-вузлів 10-19 і OR-вузлів 5-9, причому ці параметри варіюються незалежно. Це означає, що складність сценаріїв не є монотонною функцією розміру графа. Зокрема, сценарій 4 при 31 вузлі й 52 ребрах має підвищену частку OR-залежностей і щільніший простір альтернативних траєкторій, тоді як сценарій 3 при 42 вузлах і 74 ребрах характеризується більшою глибиною та домінуванням послідовних AND-залежностей, що звужує простір допустимих дій. Для кожного сценарію кібератаки виконано 30 незалежних запусків кожного методу, що забезпечує статистично репрезентативне оцінювання.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Першим кроком аналізу є оцінювання сумарного часу компрометації для зазначених підходів. Відповідні результати наведено на рис. 2а та 2б, які відображають підмножини Сценарій 1-5 (Scenario1-5) та Сценарій 6-10 (Scenario 6-10) відповідно.

Запропонований підхід демонструє найменші значення TTC у 9 із 10 сценаріїв, але характер переваги залежить від структури графа. На компактних графах перевага над shortest path природно зменшується: у сценарії 1 Proposed дає 12.46 проти 12.60, тобто метод не отримує штучного виграшу на кожному графі. Натомість у сценаріях із більшою глибиною та часткою складних залежностей виграш стає суттєвим: у сценарії 3 Proposed знижує TTC до 17.97 проти 21.25 для shortest path і 25.49 для RL без обмежень, у сценарії 5 – до 19.09 проти 21.55 і 27.15 відповідно. Отже, головний ефект моделі виникає там, де семантичні залежності реально впливають на простір доступних рішень.

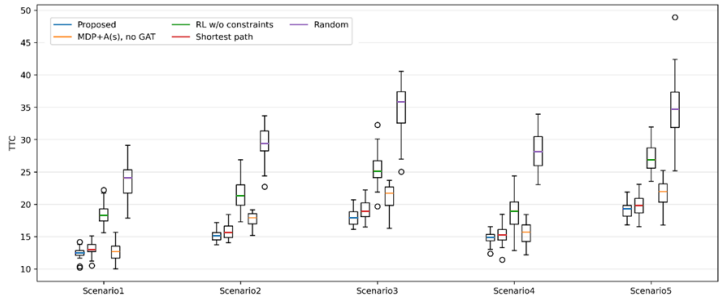

Рис. 2а. Розкид значень TTC для сценаріїв 1-5

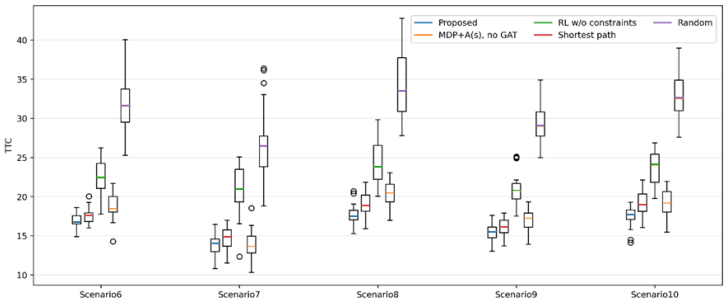

Рис. 2б. Розкид значень TTC для сценаріїв 6-10

Другим компонентом перевірки є кількість нерелевантних дій, тобто переходів, що порушують семантичні передумови MAL-графа (рис. 3). Цей показник є прямим тестом того, чи коректно модель реалізує обмеження $A(s)$.

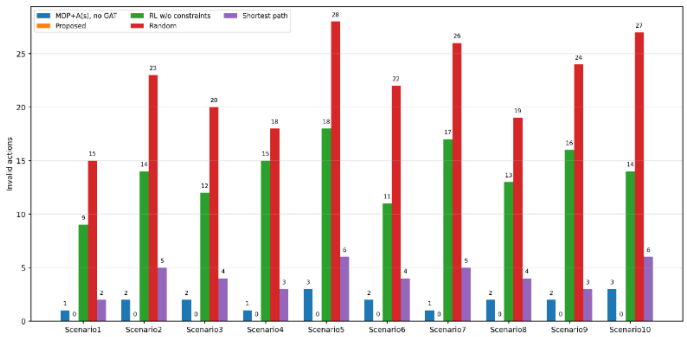

Рис. 3. Залежність кількості нерелевантних дій від методу розрахунку та сценарію

З рис. 3 видно, що для Proposed кількість нерелевантних дій дорівнює нулю в усіх десяти сценаріях, що підтверджує роботу обмеження A(s) саме як механізму побудови траєкторії, а не постфільтра. MDP_As_no_GAT дає 1-3 нерелевантні дії, тобто семантична фільтрація є визначальною, але графове представлення додатково впливає на локальну точність. Shortest path демонструє 2-6 нерелевантних дій: метод знаходить короткі траєкторії, але не гарантує їх виконуваність у динамічному просторі станів. Найгірший результат у RL без обмежень – 9-18 нерелевантних дій, причому ця величина не зростає монотонно з розміром графа (у сценарії 4 при 31 вузлі – 15, у сценарії 3 при 42 вузлах – 12), що вказує на неврахування саме семантичної структури, а не розмірності. Випадкова стратегія досягає 27-28, майже повністю втрачаючи зв'язок із логікою середовища.

Третій компонент – динаміка навчання (рис. 4): вона дозволяє оцінити не лише фінальні значення TTC, а й швидкість збіжності та стабільність процесу, що для задач RL не менш важливі за фінальну якість. На рис. 3 наведено криві лише для трьох методів – Proposed, MDP_As_no_GAT та RL без обмежень, – оскільки shortest path і випадкова стратегія не є алгоритмами машинного навчання і не мають фази навчання, а дають фіксований результат на кожному запуску.

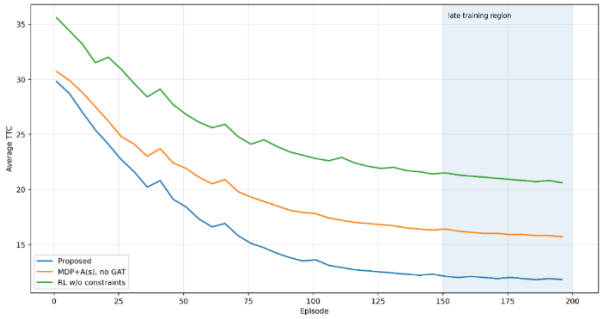

Рис. 4. Динаміка навчання для різних методів за середнім значенням TTC

Усі криві на рис. 3 мають немонотонний характер, але різко відрізняються за стабільністю. Proposed демонструє найшвидшу збіжність і стабілізується на найнижчому рівні TTC. MDP_As_no_GAT слідує тій самій траєкторії, але на систематично вищих значеннях. RL без обмежень поводиться нестабільно і завершує навчання зі значно гіршим TTC, що підтверджує важливість семантичного обмеження A(s).

Отримані результати узагальнено на рівні агрегованих метрик у табл. 1. Вона потрібна для перевірки того, чи збігаються висновки за різними критеріями одночасно. Якщо Proposed виграє лише за одним показником, але програє за іншими, такий результат не можна вважати переконливим.

Таблиця 1

Узагальнені показники ефективності методів

Метод	Середнє значення TTC	Стандартне відхилення TTC	Середня кількість кроків	Середня кількість нерелевантних дій	Частка успішних сценаріїв
Proposed_MDP_As_GAT	16.05	2.36	5.9	0.0	0.94
MDP_As_no_GAT	16.96	2.50	6.5	1.9	0.88
RL_no_constraints	22.33	3.10	8.4	13.9	0.76
Shortest_path_static_graph	17.77	3.06	6.3	4.2	0.85
Random_policy	29.61	4.66	11.1	22.2	0.37

Дані в табл. 1 свідчать, що Proposed забезпечує найкращий загальний баланс між усіма метриками. Виграш відносно RL без обмежень становить 6.28 одиниці TTC (≈28%), а відносно shortest path – 1.72 одиниці (≈9.7%), причому в обох випадках супроводжується зменшенням стандартного відхилення, що свідчить про підвищення стабільності. Порівняння Proposed і MDP_As_no_GAT показує, що GAT дає додатковий виграш 0.91 одиниці TTC при одночасному зменшенні стандартного відхилення з 2.50 до 2.36, тобто покращує не лише середнє значення, а й повторюваність результатів. Важливо зазначити, що shortest path виявляється близьким до Proposed за довжиною траєкторії (6.3 проти 5.9 кроків), але гіршим за TTC, тобто він мінімізує кількість переходів, а не очікуваний час компрометації. RL без обмежень поступається одночасно і за довжиною, і за часом, що свідчить про структурну неефективність його політики. Показники нерелевантних дій і частки успішних сценаріїв (колонки 4 і 5 табл. 1) підтверджують цю ієрархію методів у тому ж порядку.

Оскільки виграш методу має бути інтерпретований не лише на рівні агрегованих чисел, а й через властивості середовища, окремо наводиться табл. 2 для структурних характеристик сценаріїв. Вона слугує не формальним описом набору даних, а поясненням, чому ті самі методи поводяться по-різному на різних сценаріях і чому складність не зменшується до $|V|$ або $|E|$.

Структурні характеристики сценаріїв, які пояснюють різницю поведінки методів на різних графах, наведено у табл. 2.

Таблиця 2

Структурні характеристики сценаріїв кібератак

№ Сценарію	Вузли	Ребра	AND-вузли	OR-вузли	Базові вузли	Середнє TTC	Глибина
1	27	43	10	5	12	3.0	5
2	35	59	13	6	16	3.4	6
3	42	74	17	7	18	3.9	8
4	31	52	12	8	11	3.2	6
5	48	88	19	7	22	4.1	8
6	37	64	14	6	17	3.6	7
7	29	55	9	9	11	3.3	5
8	44	79	16	8	20	3.8	7
9	33	57	11	7	16	3.5	6
10	46	83	18	6	22	3.9	7

Сценарії у табл. 2 не утворюють впорядкованої шкали складності. Найбільший розрив між Proposed і базовими методами спостерігається у сценаріях 3 і 5 – з максимальною глибиною (8) і найбільшою кількістю AND-вузлів (17 і 19), – що підтверджує важливість моделювання послідовних залежностей у глибоких графах. Натомість у сценаріях із домінуванням OR-залежностей (сценарії 4 і 7) shortest path лишається конкурентним. Компактний граф з великою кількістю альтернатив дозволяє топологічно короткій траєкторії бути близькою до часово ефективної. У сценарії 10, де поєднуються значний розмір і AND-домінування, shortest path уже не витримує конкуренції, а MDP_As_no_GAT програє через менш ефективне узагальнення.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Запропоновано модель подання сценаріїв кібератак у вигляді марковського процесу прийняття рішень, у якій динаміка переходів між станами визначається виконаними кроками атаки, а множина допустимих дій формується з урахуванням семантичних залежностей, зокрема типу AND/OR. Це забезпечує часову інтерпретацію сценаріїв через метрику TTC та дозволяє формально описувати як прості, так і складні багатокрокові траєкторії компрометації.

Експериментальні результати підтверджують узгоджену перевагу підходу за кількома ключовими показниками. Середній час до компрометації становить 16.05 кроків у запропонованому підході порівняно з 22.33 для RL без обмежень (зменшення на 28%) і 17.77 для shortest path (зменшення майже на 10%), при одночасному зниженні стандартного відхилення до 2.36 проти 3.10 і 3.06 відповідно, що свідчить про підвищення стабільності. Середня довжина траєкторії скорочується до 5.9 кроків проти 8.4 для RL, що вказує на ефективніше планування. При цьому кількість нерелевантних дій у запропонованому підході відсутня порівняно з 13.9 для RL і 4.2 для RL і 4.2 для графового базового методу, що підтверджує семантичну коректність побудованих сценаріїв. Частка успішних проходжень досягає 0.94 проти 0.76 для RL і 0.85 для shortest path, що відображає підвищення надійності. Додатково встановлено, що використання графового представлення дає стабільне, хоча й помірне покращення (≈0.9-1.4 одиниці TTC), тоді як максимальний виграш моделі спостерігається у сценаріях із глибиною 8 та кількістю AND-вузлів 17-19, де різниця TTC досягає 6-8 одиниць.

Практичне значення полягає в можливості застосування моделі для кількісного оцінювання часу компрометації, ранжування альтернативних сценаріїв атак, виявлення критичних вузлів інфраструктури та підтримки прийняття рішень у задачах аналізу безпеки, зокрема в системах автоматизованого тестування та навчальних кіберполігонах.

Обмеження дослідження пов'язані з використанням стохастично-згенерованих, хоча й структурно узгоджених із відкритими джерелами, MAL-графів, обмеженим набором базових методів без включення сучасних алгоритмів глибинного навчання з підкріпленням, а також спрощенням моделі переходів і використанням очікуваних значень TTC без урахування повної стохастики виконання атак.

Подальші дослідження доцільно спрямувати на розширення моделі до багатоагентної постановки «атака–захист», інтеграцію більш складних алгоритмів навчання з підкріпленням, а також побудову теоретичних оцінок властивостей отриманих політик і дослідження масштабованості моделі. Окремий напрям передбачає дослідження відповідності запропонованого підходу міжнародним стандартам і методологіям тестування на проникнення, зокрема OWASP, PTES, NIST SP 800-115 та OSSTMM [22], а також його узгодження з моделями опису тактик і технік атак MITRE ATT&CK. Такий аналіз дозволить визначити, наскільки згенеровані агентом траєкторії компрометації відповідають реальним сценаріям,

що використовуються у практиці тестування безпеки, і створить передумови для інтеграції методу в наявні інструменти автоматизованого тестування веб-додатків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zenitani, K. (2023). Attack graph analysis: An explanatory guide. *Computers & Security, 126*, 103081. https://doi.org/10.1016/j.cose.2022.103081
2. Ibrahim, M., & Elhafiz, R. (2022). Integrated clinical environment security analysis using reinforcement learning. *Bioengineering, 9*(6), 253. https://doi.org/10.3390/bioengineering9060253
3. Kaya, M. O., Ozdem, M., & Das, R. (2025). A new hybrid approach combining GCN and LSTM for real-time anomaly detection from dynamic computer network data. *Computer Networks, 268*, 111372. https://doi.org/10.1016/j.comnet.2025.111372
4. Xie, R., & Liu, D. (2026). A novel hybrid graph neural network and transformer model for intrusion detection. *Peer-to-Peer Networking and Applications,19*(2). https://doi.org/10.1007/s12083-025-02171-w
5. Vitulyova, Y., Babenko, T., Kolesnikova, K., Kiktev, N., & Abramkina, O. (2025). A hybrid approach using graph neural networks and LSTM for attack vector reconstruction. *Computers, 14*(8), 301. https://doi.org/10.3390/computers14080301
6. Yousefi, M., Mtetwa, N., Zhang, Y., & Tianfield, H. (2018). A reinforcement learning approach for attack graph analysis. In *2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications / 12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)* (pp. 212-217). IEEE. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00041
7. Yu, Z., Jia, Y., Han, W., Zhang, J., Yang, M., & Mei, Y. (2025). ShotFlex: A reinforcement learning-based cyber attack path generation method for cybersecurity evaluation. *Security and Safety, 4*, 2025006. https://doi.org/10.1051/sands/2025006
8. Kim, B.-S., Suk, H.-W., Choi, Y.-H., Moon, D.-S., & Kim, M.-S. (2024). Optimal cyber attack strategy using reinforcement learning based on Common Vulnerability Scoring System. *Computer Modeling in Engineering & Sciences, 141*(2), 1551-1574. https://doi.org/10.32604/cmes.2024.052375
9. Abdullayeva, F., & Suleymanzade, S. (2024). Cyber security attack recognition on cloud computing networks based on graph convolutional neural network and GraphSAGE models. *Results in Control and Optimization, 15*, 100423. https://doi.org/10.1016/j.rico.2024.100423
10. Ren, W., Zhang, H., & Lei, Y. (2025). Network attack knowledge inference with graph convolutional networks and convolutional 2D KG embeddings. *Scientific Reports, 15*(1). https://doi.org/10.1038/s41598-025-17941-y
11. Liu, G., Lu, K., & Pi, S. (2025). Graph neural networks embedded with domain knowledge for cyber threat intelligence entity and relationship mining. *PeerJ Computer Science, 11*, e2769. https://doi.org/10.7717/peerj-cs.2769
12. Li, Y., & Li, X. (2021). Research on multi-target network security assessment with attack graph expert system model. *Scientific Programming, 2021*, 1-11. https://doi.org/10.1155/2021/9921731
13. Levner, E., & Tsadikovich, D. (2024). Fast algorithm for cyber-attack estimation and attack path extraction using attack graphs with AND/OR nodes. *Algorithms, 17*(11), 504. https://doi.org/10.3390/a17110504
14. Ibrahim, A., Bozhinoski, S., & Pretschner, A. (2019). Attack graph generation for microservice architecture. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* (pp. 1235-1242). ACM. https://doi.org/10.1145/3297280.3297401
15. Prytula, A., & Kupershtein, L. (2025). Analysis of penetration testing approaches using reinforcement learning. *Cybersecurity: Education, Science, Technique, 4*(28), 259-271. https://doi.org/10.28925/2663-4023.2025.28.789
16. Johnson, P., Lagerström, R., & Ekstedt, M. (2018). A meta language for threat modeling and attack simulations. In *Proceedings of the 13th International Conference on Availability, Reliability and Security* (pp. 1-8). ACM. https://doi.org/10.1145/3230833.3232799
17. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations (ICLR 2018)*. https://doi.org/10.48550/arXiv.1710.10903
18. mal-lang. (n.d.). *enterpriseLang: Enterprise language for the Meta Attack Language framework* [Software]. GitHub. https://github.com/mal-lang/enterpriseLang
19. OWASP Foundation. (n.d.). *WebGoat: A deliberately insecure web application* [Software]. GitHub. https://github.com/WebGoat/WebGoat

20. Torrano-Gimenez, C., Perez-Villegas, A., & Alvarez, G. (2010). *HTTP Dataset CSIC 2010* [Dataset]. Spanish National Research Council (CSIC). https://www.kaggle.com/datasets/ispangler/csic-2010-web-application-attacks
21. Kaggle. (n.d.). *Malicious URL Detection Dataset* [Dataset]. Kaggle. https://www.kaggle.com/datasets/moutasmtamimi/malicious-url-detection-dataset-enhanced-2026
22. Kupershtein, L. M., Prytula, A. V., & Malinovskyi, V. I. (2024). Analysis of web applications penetration testing technologies. *Scientific Works of Vinnytsia National Technical University*, *2*, 45-53. https://doi.org/10.31649/2307-5376-2024-2-45-53

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (33), 2026
ISSN 2663 – 4023

Andrii Prytula
PhD student of Information Protection Department
Vinnytsia national technical University, Vinnytsia, Ukraine
ORCID: 0009-0006-9632-0712
andrik.pritula@gmail.com

Leonid Kupershtein
PhD, Associate Professor of Information Protection Department
Vinnytsia national technical University, Vinnytsia, Ukraine
ORCID: 0000-0001-6737-7134
kupershtein.lm@gmail.com

MODELING CYBERATTACK SCENARIOS AS A MARKOV DECISION PROCESS WITH A SEMANTICALLY CONSTRAINED ACTION SPACE

Abstract. A formal model for representing cyberattack scenarios as a Markov decision process is proposed, in which, unlike static attack graphs, the dynamics of system state changes depending on the executed attack steps are explicitly defined, while the set of admissible actions is formed considering semantic dependencies between steps, in particular AND and OR type dependencies. The proposed approach provides a temporal interpretation of scenarios through the time-to-compromise (TTC) metric and allows describing both simple and complex multi-step compromise trajectories. The model combines a dynamic MDP representation with an invariant graph representation of states, constructed using graph neural network mechanisms. The experimental study was conducted on a set of stochastically generated MAL-graphs aligned with open attack models and web datasets and includes a comparison with baseline graph-based methods and reinforcement learning methods without semantic constraints. The obtained results show that the proposed approach provides a substantial reduction of the average time to compromise and decreases the variance of results, which indicates improved learning stability. It is demonstrated that the introduction of a semantically constrained action set eliminates irrelevant transitions and significantly increases the share of successful compromise scenarios. The greatest gain is observed on deep multi-step attack trajectories dominated by AND dependencies, where the semantic structure of the graph has a decisive impact on the space of available decisions. The practical significance lies in the possibility of applying the model for quantitative evaluation of cyberattack scenarios, ranking of compromise trajectories and decision support, as well as integration into automated penetration testing systems and cyber training ranges.

Keywords: markov process; decision making; graph; cybersecurity; threat; reinforcement learning; neural network; modeling; machine learning; artificial intelligence

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Zenitani, K. (2023). Attack graph analysis: An explanatory guide. *Computers & Security*, *126*, 103081. https://doi.org/10.1016/j.cose.2022.103081
2. Ibrahim, M., & Elhafiz, R. (2022). Integrated clinical environment security analysis using reinforcement learning. *Bioengineering*, *9*(6), 253. https://doi.org/10.3390/bioengineering9060253
3. Kaya, M. O., Ozdem, M., & Das, R. (2025). A new hybrid approach combining GCN and LSTM for real-time anomaly detection from dynamic computer network data. *Computer Networks*, *268*, 111372. https://doi.org/10.1016/j.comnet.2025.111372
4. Xie, R., & Liu, D. (2026). A novel hybrid graph neural network and transformer model for intrusion detection. *Peer-to-Peer Networking and Applications*,*19*(2). https://doi.org/10.1007/s12083-025-02171-w
5. Vitulyova, Y., Babenko, T., Kolesnikova, K., Kiktev, N., & Abramkina, O. (2025). A hybrid approach using graph neural networks and LSTM for attack vector reconstruction. *Computers*, *14*(8), 301. https://doi.org/10.3390/computers14080301
6. Yousefi, M., Mtetwa, N., Zhang, Y., & Tianfield, H. (2018). A reinforcement learning approach for attack graph analysis. In *2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications / 12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)* (pp. 212-217). IEEE. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00041

7. Yu, Z., Jia, Y., Han, W., Zhang, J., Yang, M., & Mei, Y. (2025). ShotFlex: A reinforcement learning-based cyber attack path generation method for cybersecurity evaluation. *Security and Safety*, *4*, 2025006. https://doi.org/10.1051/sands/2025006
8. Kim, B.-S., Suk, H.-W., Choi, Y.-H., Moon, D.-S., & Kim, M.-S. (2024). Optimal cyber attack strategy using reinforcement learning based on Common Vulnerability Scoring System. *Computer Modeling in Engineering & Sciences*, *141*(2), 1551-1574. https://doi.org/10.32604/cmes.2024.052375
9. Abdullayeva, F., & Suleymanzade, S. (2024). Cyber security attack recognition on cloud computing networks based on graph convolutional neural network and GraphSAGE models. *Results in Control and Optimization*, *15*, 100423. https://doi.org/10.1016/j.rico.2024.100423
10. Ren, W., Zhang, H., & Lei, Y. (2025). Network attack knowledge inference with graph convolutional networks and convolutional 2D KG embeddings. *Scientific Reports*, *15*(1). https://doi.org/10.1038/s41598-025-17941-y
11. Liu, G., Lu, K., & Pi, S. (2025). Graph neural networks embedded with domain knowledge for cyber threat intelligence entity and relationship mining. *PeerJ Computer Science*, *11*, e2769. https://doi.org/10.7717/peerj-cs.2769
12. Li, Y., & Li, X. (2021). Research on multi-target network security assessment with attack graph expert system model. *Scientific Programming*, *2021*, 1-11. https://doi.org/10.1155/2021/9921731
13. Levner, E., & Tsadikovich, D. (2024). Fast algorithm for cyber-attack estimation and attack path extraction using attack graphs with AND/OR nodes. *Algorithms*, *17*(11), 504. https://doi.org/10.3390/a17110504
14. Ibrahim, A., Bozhinoski, S., & Pretschner, A. (2019). Attack graph generation for microservice architecture. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* (pp. 1235-1242). ACM. https://doi.org/10.1145/3297280.3297401
15. Prytula, A., & Kupershtein, L. (2025). Analysis of penetration testing approaches using reinforcement learning. *Cybersecurity: Education, Science, Technique*, *4*(28), 259-271. https://doi.org/10.28925/2663-4023.2025.28.789
16. Johnson, P., Lagerström, R., & Ekstedt, M. (2018). A meta language for threat modeling and attack simulations. In *Proceedings of the 13th International Conference on Availability, Reliability and Security* (pp. 1-8). ACM. https://doi.org/10.1145/3230833.3232799
17. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations (ICLR 2018)*. https://doi.org/10.48550/arXiv.1710.10903
18. mal-lang. (n.d.). *enterpriseLang: Enterprise language for the Meta Attack Language framework* [Software]. GitHub. https://github.com/mal-lang/enterpriseLang
19. OWASP Foundation. (n.d.). *WebGoat: A deliberately insecure web application* [Software]. GitHub. https://github.com/WebGoat/WebGoat
20. Torrano-Gimenez, C., Perez-Villegas, A., & Alvarez, G. (2010). *HTTP Dataset CSIC 2010* [Dataset]. Spanish National Research Council (CSIC). https://www.kaggle.com/datasets/ispangler/csic-2010-web-application-attacks
21. Kaggle. (n.d.). *Malicious URL Detection Dataset* [Dataset]. Kaggle. https://www.kaggle.com/datasets/moutasmtamimi/malicious-url-detection-dataset-enhanced-2026
22. Kupershtein, L. M., Prytula, A. V., & Malinovskyi, V. I. (2024). Analysis of web applications penetration testing technologies. *Scientific Works of Vinnytsia National Technical University*, *2*, 45-53. https://doi.org/10.31649/2307-5376-2-45-53

Отримано редакцією журналу / Received: 23.02.26
Прорецензовано / Revised: 02.03.26
Схвалено до друку / Accepted: 25.06.26