https://doi.org/10.31891/2219-9365-2026-87-14
УДК 004.056+004.85

ПРИТУЛА Андрій
Вінницький національний технічний університет
https://orcid.org/0009-0006-9632-0712
e-mail: andrik.pritula@gmail.com
КУПЕРШТЕЙН Леонід
Вінницький національний технічний університет
https://orcid.org/0000-0001-6737-7134
e-mail: kupershtein.lm@gmail.com

СТОХАСТИЧНА МОДЕЛЬ ПРОТИБОРСТВА «АТАКА–ЗАХИСТ» ДЛЯ ТЕСТУВАННЯ БЕЗПЕКИ ВЕБЗАСТОСУНКІВ НА ОСНОВІ НАВЧАННЯ З ПІДКРІПЛЕННЯМ

У статті запропоновано стохастичну модель протиборства «атака–захист» для тестування безпеки вебзастосунків на основі навчання з підкріпленням. Актуальність роботи зумовлена зростанням складності сучасних вебзастосунків і адаптивним характером кіберзагроз, для яких статичні графи атак і односторонні моделі тестування не враховують активної реакції захисту та часової динаміки компрометації. Проведений аналіз наявних підходів показав, що більшість моделей розглядає лише атакувальну або лише захисну сторону, не використовує час до компрометації як основний критерій оптимізації та формує надмірно великі простори станів, що ускладнює аналіз адаптивних сценаріїв. На відміну від односторонніх моделей, підхід реалізовано у вигляді двоагентної марковської гри, у якій одночасно враховуються дії агента тестування та захисного агента в спільному просторі станів із двобічною функцією винагороди, орієнтованою на час до компрометації, а політика навчається ризик-чутливим методом навчання з підкріпленням, що спрямовує пошук на короткі критичні траєкторії. Модель використовує агреговане подання системних компонентів як агрегацію графа атак, побудованого засобами Meta Attack Language, і стохастичне задання початкових умов, що дозволяє зменшити розмірність простору станів і забезпечити обчислювальну керованість аналізу. Результати емпіричної перевірки на наборах даних ToN-IoT і CIC-IDS2017 показали, що запропонована модель скорочує середній час до компрометації на 34,7–37,7% порівняно з класичними графами атак, а розмірність фактично відвіданого простору станів зменшує у 8–12 разів, що пришвидшує аналіз сценаріїв і поліпшує масштабованість. Активна поведінка захисту дозволяє оцінювати не лише факт, а й часові характеристики компрометації. Отримані результати доцільно використовувати як формальну основу адаптивних засобів тестування безпеки вебзастосунків у середовищах із частковою спостережуваністю та змінними параметрами захисту.

Ключові слова: тестування безпеки вебзастосунків, навчання з підкріпленням, марковська гра, час до компрометації, двоагентна модель, стохастичне моделювання, кібербезпека, машинне навчання, кібератака, загроза, штучний інтелект.

PRYTULA Andrii, KUPERSHTEIN Leonid
Vinnytsia National Technical University

STOCHASTIC ATTACK–DEFENSE CONFRONTATION MODEL FOR WEB APPLICATION SECURITY TESTING BASED ON REINFORCEMENT LEARNING

The paper proposes a stochastic attack–defense confrontation model for web application security testing based on reinforcement learning. The relevance of the work stems from the growing complexity of modern web applications and the adaptive nature of cyber threats, for which static attack graphs and one-sided testing models fail to account for active defense responses and the temporal dynamics of compromise. An analysis of existing approaches showed that most models consider only the attacking or only the defending side, do not use time-to-compromise as the main optimization criterion, and generate excessively large state spaces that hinder the analysis of adaptive scenarios. Unlike one-sided models, the approach is implemented as a two-agent Markov game, in which the actions of the testing agent and the defense agent are considered simultaneously in a shared state space with a two-sided reward function oriented toward time-to-compromise (TTC), while the policy is trained using a risk-sensitive reinforcement learning method that directs the search toward short critical trajectories. The model uses an aggregated representation of system components as an aggregation of an attack graph built with the Meta Attack Language, together with a stochastic initialization of initial conditions, which reduces the dimensionality of the state space and ensures the computational tractability of the analysis. Empirical evaluation on the ToN-IoT and CIC-IDS2017 datasets showed that the proposed model reduces the average time-to-compromise by 34.7–37.7 % compared to classical attack graphs and decreases the dimensionality of the actually visited state space by a factor of 8–12, which accelerates scenario analysis and improves scalability. The active behavior of the defense makes it possible to evaluate not only the fact but also the temporal characteristics of compromise. The results can be used as a formal basis for adaptive web application security testing tools in environments with partial observability and variable defense parameters.

Keywords: web application security testing, reinforcement learning, Markov game, time to compromise, two-agent model, stochastic modeling, cybersecurity, machine learning, cyberattack, threat, artificial intelligence.

Стаття надійшла до редакції / Received 23.03.2026
Прийнята до друку / Accepted 02.07.2026
Опубліковано / Published 10.09.2026

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ЗІ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Стрімке зростання складності сучасних вебзастосунків, поява багаторівневих хмарних архітектур і збільшення кількості взаємопов'язаних вебсервісів суттєво ускладнюють задачі тестування безпеки та аналізу сценаріїв компрометації. Сучасні кіберзагрози дедалі частіше мають адаптивний характер, змінюють траєкторію залежно від реакції захисного агента та реалізуються як послідовність взаємопов'язаних дій у середовищах із частковою спостережуваністю. У зв'язку з цим статичні графи атак і детерміновані сценарії тестування виявляються недостатніми для аналізу динамічного кіберпротиборства [1]. Одночасно огляд сучасних підходів до використання навчання з підкріпленням у мережевій безпеці демонструє, що адаптивні моделі кіберзахисту стають одним із ключових напрямків розвитку інтелектуальних систем безпеки [2].

Одним із базових напрямків автоматизації тестування безпеки є використання графів атак для аналізу можливих траєкторій компрометації. Класичний підхід до побудови графа атак було закладено у праці [3], де граф атак використовувався для формального опису досяжності вразливих станів мережі. Подальший розвиток цього напряму представлено в роботі [4], у якій графи сценаріїв застосовано для формалізації багатокрокових атак. Проте такі моделі мають переважно статичний характер і не враховують адаптивної реакції захисного агента. У статті [5] імовірнісний граф атак поєднується з глибоким навчанням з підкріпленням для оптимізації адаптивного захисту зі зміною конфігурації цілі, однак основна увага приділяється оборонній політиці, а не задачі тестування безпеки вебзастосунків. Аналогічно у праці [6] граф атак використовується для аналізу критичних вузлів мережі, але система захисту залишається фактично статичним елементом середовища.

Другий напрямок досліджень пов'язаний із використанням одноагентного навчання з підкріпленням для автоматизації аналізу графів атак і тестування на проникнення. У роботі [7] запропоновано використання Q-навчання для кількісного оцінювання безпеки мереж на основі графів атак. У статті [8] навчання з підкріпленням використовується для пошуку оптимальних шляхів ексфільтрації даних у корпоративних мережах. У праці [9] розглядаються проблеми формалізації середовищ для автоматизованого тестування на проникнення. У роботі [10] глибоке навчання з підкріпленням використовується для моделювання кібератак у реалістичних сценаріях. Незважаючи на перспективність цих підходів, вони моделюють переважно поведінку атакувального агента й не враховують повноцінної адаптивної взаємодії між атакою та захистом у спільному просторі станів.

Окремий напрямок становлять частково спостережувані, ігрові та багатоагентні моделі кіберпротиборства. У статті [11] запропоновано модель частково спостережуваного марковського процесу прийняття рішень адаптивного кіберзахисту для багатокрокових атак, проте вона зосереджена на побудові захисної політики, а не на тестуванні безпеки вебзастосунків. У роботі [12] використано марковські моделі рухомої цільової оборони, але без орієнтації на часові характеристики компрометації. У праці [13] поєднано ігрові методи й машинне навчання для побудови адаптивних стратегій кіберзахисту, проте модель орієнтована переважно на стратегічний рівень прийняття рішень. У статті [14] запропоновано марковську гру між атакувальником і захисником у хмарному середовищі, однак модель не враховує специфіку тестування безпеки вебзастосунків і не використовує часові характеристики компрометації як цільовий критерій оптимізації.

Проведений аналіз показує, що сучасні підходи мають щонайменше три суттєві обмеження. По-перше, значна частина моделей розглядає лише атакувальника або лише захисника без повноцінного моделювання двобічного протиборства. По-друге, часові характеристики компрометації фактично не використовуються як основний критерій оптимізації сценаріїв тестування. По-третє, деталізовані графові моделі формують надмірно великі простори станів, що істотно ускладнює аналіз адаптивних сценаріїв і знижує обчислювальну керованість моделей.

Зазначені обмеження частково розв'язано в попередніх роботах авторів. У роботі [15] запропоновано формалізацію сценаріїв кібератак як марковського процесу прийняття рішень із семантично обмеженим простором допустимих дій на основі AND/OR-залежностей графа атак і часовою інтерпретацією траєкторій через метрику TTC. У роботі [16] цю постановку розширено функцією винагороди, що поєднує часову складову та семантичну важливість проміжних цілей атаки. Однак обидві моделі є односторонніми. Захисний агент в них не виступає активним агентом, а час до компрометації розглядається за пасивного захисту. Поточне дослідження усуває це обмеження, розширюючи односторонню постановку до двоагентної марковської гри, у якій тестовий вплив і захисна реакція формуються одночасно у спільному просторі станів.

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Метою дослідження є підвищення ефективності тестування безпеки вебзастосунків шляхом підвищення здатності моделі виявляти критичні сценарії з малим часом до компрометації та одночасного зменшення розмірності простору станів під час аналізу адаптивних сценаріїв кіберпротиборства.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Модель стохастичного тестування безпеки вебзастосунків

Класичні графові моделі атак описують лише структурну досяжність цільових компонентів і не враховують одночасну зміну тестового впливу, захисних параметрів, видимості вузлів і часу до досягнення вразливого стану. Натомість у реальному середовищі вебзастосунку тестова дія змінює інформаційний стан агента, впливає на подальший вибір дій і може активувати механізми виявлення, фільтрації або блокування.

Структурною основою моделі залишається граф атак, побудований засобами метамови Meta Attack Language (MAL) [15], у якому вузли відповідають крокам атаки над активами вебзастосунку, а семантичні залежності між кроками задаються композиціями типу AND та OR. Однак для двобічного протиборства «атака–захист» подання на рівні окремих кроків атаки є надлишково деталізованим, оскільки захисні дії та моніторинг природно визначаються не для кожного кроку, а для активів, які ці кроки експлуатують. Тому MAL-граф агрегується до рівня функціональних компонентів. Множина кроків атаки над одним активом згортається в його атрибути тестового впливу, а захисні властивості активу задаються оборонними атрибутами та параметром моніторингу. Саме ця агрегація забезпечує перехід від покрокової семантики MAL до керованого простору станів двоагентної гри без втрати типології активів і дій, успадкованої з MAL-специфікації.

Вебзастосунок подається у вигляді графа $G=\left(N,E\right)$, де N є множиною вебкомпонентів, зокрема API-вузлів, модулів автентифікації, серверних обробників, баз даних і сервісів керування сесіями тощо, а $E\subseteq N\times N$ задає функціональні та топологічні зв'язки між ними. Ребро $(N_i,N_j)\in E$ означає можливість переходу між компонентами під час виконання сценарію тестування. У такому поданні маршрут у графі задає лише потенційну послідовність дій, результат якої залежить від стану компонента, рівня захисту, імовірності виявлення та реакції захисного агента, а не гарантує компрометацію.

Процес тестування безпеки формалізується як двоагентна марковська гра $\Gamma=(S,A^{test},A^{def},P,R^{test},R^{def})$, де S є спільним простором станів веборієнтованої інформаційно-комунікаційної системи, A^{test} є множиною дій агента тестування безпеки, A^{def} є множиною дій захисного агента, P є стохастичним ядром переходів. Функція R^{test} задає винагороду агента тестування й заохочує швидке досягнення вразливого стану за низької ймовірності виявлення, тоді як R^{def} задає винагороду захисного агента й заохочує раннє виявлення тестової активності та підвищення часової стійкості компонентів. Функції винагороди є різноспрямованими.

У момент часу t система перебуває у стані $s_t\in S$. Після одночасного вибору дій $a_t^{test}\in A^{test}$ і $a_t^{def}\in A^{def}$ формується наступний стан s_{t+1}. Імовірність переходу визначається як $\Pr(s_{t+1}=s'\mid s_t=s,a_t^{test},a_t^{def})=P(s'\mid s,a_t^{test},a_t^{def})$. Марковська властивість полягає в тому, що наступний стан визначається поточним станом і парою дій агентів, а не всією попередньою історією сценарію. Загальну структуру взаємодії агентів у запропонованій моделі показано на рис. 1. Зі спільного стану агент тестування формує дію на основі частково спостережуваної оцінки оборонних атрибутів, тоді як захисний агент діє за повною інформацією про власний стан; обидві дії одночасно надходять до ядра переходу, яке формує наступний стан, накопичує час до компрометації та повертає винагороди обом агентам.

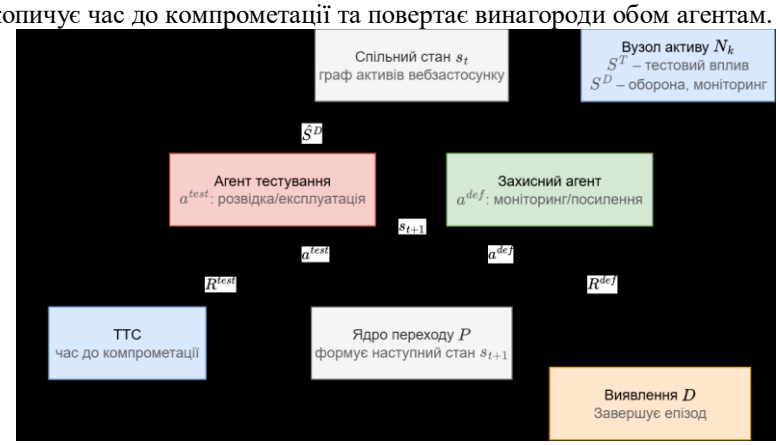

Рис.1 Схема двоагентної марковської гри протиборства «атака-захист»

Локальний стан вебкомпонента $N_k\in N$ задається парою $S_k=(S_k^T,S_k^D)$, де S_k^T є вектором атакуючого впливу, а S_k^D є вектором оборонних атрибутів. Компонента $S_{k,j}^T$ описує накопичений ефект тестових дій типу j щодо компонента N_k. У контексті вебзастосунку такий ефект може відповідати

просуванню сценарію ін'єкції, обходу автентифікації, зловживанню API-викликами, підвищенню рівня доступу або компрометації сесійного механізму. Компонента $S_{k,j}^{D}$ задає рівень протидії відповідному типу тестового впливу, зокрема фільтрацію вхідних даних, контроль доступу, перевірку токенів, обмеження частоти запитів або політику журналювання. Компонента $S_{k,j}^{D}$ також описує здатність вузла до виявлення підозрілої активності. Усі атрибути змінюються в дискретній шкалі від 0 до w, де w є максимальним рівнем атрибута.

Досягнення вразливого стану компонента за типом тестової дії j визначається умовою

$$S_{k,j}^{T} > S_{k,j}^{D}. \tag{1}$$

Вираз (1) не ототожнює вразливість із наявністю ребра у графі. Тут фіксується момент, коли накопичений тестовий вплив перевищує поточний рівень захисного механізму. Саме тому модель відображає не статичну досяжність, а керований стохастичний процес тестування безпеки.

Часову результативність сценарію задає метрика часу до компрометації (TTC) $TTC = \min\left\{t \geq 0 : N_{target} \ досягнув\, вразливого\, стану\, у\ s_t\right\}$, що оцінює часову стійкість вебзастосунку до адаптивних тестових дій. Метрика TTC визначається як найменша кількість кроків, за яку агент тестування досягає вразливого стану цільового компонента. Мінімум береться за всіма моментами часу епізоду, у які досягнуто умову компрометації.

Імовірність виявлення тестової активості на вузлі N_k визначається через параметр моніторингу $p_k^{det} = \dfrac{S_{k,m+1}^{D}+1}{w+1}$, де w – максимальний рівень атрибута, що обмежує зростання накопиченого впливу та захисту. Зростання $S_{k,m+1}^{D}$ підвищує імовірність виявлення та збільшує очікувану вартість агресивного тестового сценарію.

Для забезпечення обчислювальної керованості модель реалізується на агрегованому рівні системних компонентів. Компонент N_k репрезентує функціонально значущий елемент вебзастосунку, а не окремий HTTP-запит, системний виклик або мережевий пакет, що дає верхню оцінку розмірності простору станів $|S| \leq (w+1)^{(2m+1)|N|}$. Ця оцінка ґрунтується на тому, що кожен з $|N|$ вебкомпонентів характеризується $2m+1$ атрибутами, m тестовими, m оборонними та одним атрибутом моніторингу, кожен з яких набуває значень зі шкали з $w+1$ рівнів. Оскільки стан системи є комбінацією значень усіх атрибутів усіх компонентів, загальна кількість конфігурацій не перевищує $(w+1)^{((2m+1)|N|)}$.

Формалізація тестових і захисних дій

Множина дій агента тестування у стані s_t складається з розвідувальних та експлуатаційних дій $A_t^{test}(s) = A_t^{rec}(s) \cup A_t^{exp}(s)$. Розвідувальні дії уточнюють видимість вебкомпонентів, параметри доступу, відкриті API-виклики та ознаки захисної конфігурації. Експлуатаційні дії збільшують тестовий вплив на вибраний компонент і відповідають активним перевіркам безпеки, зокрема спробам ін'єкції, обходу автентифікації, перевірці контролю доступу або тестуванню стійкості сесійного механізму.

Множина дій захисного агента визначається як $A_t^{def}(s) = A_t^{mon}(s) \cup A_t^{hard}(s)$, де $A_t^{mon}(s)$ відповідає посиленню моніторингу, а $A_t^{hard}(s)$ відповідає підвищенню захисного рівня вебкомпонента. Моніторинг збільшує імовірність виявлення тестової активості, тоді як посилення захисту підвищує поріг досягнення вразливого стану.

Часткова спостережуваність агента тестування задається через оцінку оборонного атрибута $\hat{S}_{k,j}^{D} = S_{k,j}^{D} + \varepsilon_{k,j}$, $\varepsilon_{k,j} \sim \mathrm{N}(0, \sigma^2)$. Параметр σ^2 є дисперсією шуму спостереження. За $\sigma^2 = 0$ агент тестування отримує точну інформацію про оборонний профіль компонента. За $\sigma^2 > 0$ вибір дії ґрунтується на спотвореній оцінці, що відтворює реальні умови тестування, коли конфігурація захисних механізмів є лише частково доступною. Оновлення тестового впливу, оборонного атрибута та параметра моніторингу описується системою правил:

$$\begin{cases} S_{k,j}^{T}(t+1) = \min\left\{S_{k,j}^{T}(t) + 1(a_t^{test} = test(k,j)), w\right\} \\ S_{k,j}^{D}(t+1) = \min\left\{S_{k,j}^{D}(t) + 1(a_t^{def} = defend(k,j)), w\right\} \\ S_{k,m+1}^{D}(t+1) = \min\left\{S_{k,m+1}^{D}(t) + 1(a_t^{def} = monitor(k)), w\right\} \end{cases}. \tag{2}$$

Наведена система правил (2) описує асиметрію дій агентів. Агент тестування накопичує вплив у напрямі вразливого стану, тоді як захисний агент або підвищує поріг захисту, або збільшує імовірність виявлення. Саме це переводить тестування з односторонньої процедури в двобічний адаптивний процес.

Стохастичне виявлення тестової активності моделюється бернуллівською випадковою величиною $D_{k,t} \sim Bernoulli(p_k^{det})$. Якщо $D_{k,t} = 1$, тестовий сценарій вважається виявленим захисним агентом, а епізод завершується. Якщо $D_{k,t} = 0$, агент тестування продовжує сценарій відповідно до поточного стану, доступної видимості та оцінених оборонних атрибутів.

Для узгодження локальних тестових дій із глобальною часовою метою вводиться потенціальна функція $\Phi : S \to R_{\geq 0}$, де $\Phi(s)$ задає оцінку очікуваного часу до досягнення вразливого стану з поточного стану s. Локальна зміна потенціалу визначається як $\Delta\Phi_t = \Phi(s_t) - \Phi(s_{t+1})$. Якщо $\Delta\Phi_t < 0$, наступний стан збільшує близькість до швидкого досягнення вразливого стану. Якщо $\Delta\Phi_t > 0$, захисні дії підвищують часову стійкість вебзастосунку. Такий механізм дозволяє використовувати часову інформацію ще до завершення епізоду, що є важливим для навчання з підкріпленням у середовищах із відкладеною термінальною винагородою.

Функція винагороди агента тестування визначається як

$$R_t^{test} = \lambda_c comp(s_{t+1}) - \lambda_d det(s_{t+1}) - \lambda_\Phi \Delta\Phi_t . \quad (3)$$

У формулі (3) $comp(s_{t+1})$ є індикатором досягнення вразливого стану цільового компонента, $det(s_{t+1})$ є індикатором виявлення тестового сценарію, λ_c є вагою успішного досягнення вразливого стану, λ_d є штрафом за виявлення, а λ_Φ є вагою часової зміни потенціалу. Останній член функції винагороди спрямовує політику агента до сценаріїв, які зменшують очікуваний час до компрометації.

Функція винагороди захисного агента має вигляд:

$$R_t^{def} = -\lambda_c comp(s_{t+1}) + \lambda_d det(s_{t+1}) + \lambda_\Phi \Delta\Phi_t - \lambda_r C(a_t^{def}) , \quad (4)$$

де $C(a_t^{def})$ є ресурсною вартістю оборонної дії, а λ_r є вагою ресурсних витрат. Захисний агент отримує додатну винагороду за виявлення та збільшення часової стійкості, але штрафується за надмірне використання ресурсу.

Захисний агент діє в межах ресурсного обмеження $\sum_{t=0}^{T} C(a_t^{def}) \leq B$, яке запобігає виродженню моделі в необмежене посилення всіх вузлів. За наявності бюджету *B* захисний агент має розподіляти ресурс між моніторингом, посиленням захисту критичних вузлів і збереженням ресурсу для наступних кроків.

Формалізація задає модель тестування безпеки як двоагентну марковську гру у спільному просторі станів, що одночасно враховує дії агента тестування та захисного агента, стохастичне виявлення, часткову спостережуваність і часову характеристику компрометації. Це створює формальну основу для побудови адаптивних політик на основі навчання з підкріпленням.

Політика агента тестування навчається методом глибокого Q-навчання, адаптованим до двоагентного середовища з частковою спостережуваністю. Для кожної пари «стан–дія» вводиться функція цінності дії $Q_{test}(s, a^{test})$, що оцінює очікувану сумарну винагороду агента тестування за вибору дії a^{test} у стані s з подальшим дотриманням політики. На відміну від ризик-нейтрального підходу, який оптимізує лише середнє значення винагороди, у запропонованій моделі використовується ризик-чутливий критерій, що враховує несприятливі траєкторії з аномально швидкою компрометацією. Замість оператора максимуму в цільовому значенні застосовується ентропійне усереднення Q-значень на множині допустимих дій агента тестування в наступному стані:

$$V_\beta(s') = \frac{1}{\beta} \log \sum_{a^{test} \in A^{test}(s')} \exp(\beta Q^{test}(s', a^{test})) , \quad (5)$$

де β є параметром ризик-чутливості, а s' – наступний стан. За $\beta < 0$ оператор зміщується до м'якого мінімуму й формує несхильну до ризику політику, що уникає траєкторій із критично малим часом до компрометації, тоді як за $\beta \to 0$ він переходить у математичне сподівання, відновлюючи ризик-нейтральний випадок.

Відповідно, цільове значення для оновлення Q-функції набуває вигляду

$$y_t = R_t^{test} + \gamma V_\beta(s_{t+1}) , \quad (6)$$

де R_t^{test} – миттєва винагорода агента тестування (3), а γ – коефіцієнт дисконтування. Оператор V_β замінює звичайний максимум $\max_a Q(s',a)$ у цілі Q-навчання, за $\beta<0$ він зміщується до мінімуму, формуючи несхильну до ризику оцінку наступного стану, а за $\beta\to 0$ – до математичного сподівання, що відповідає ризик-нейтральному випадку. Параметри Q-мережі θ оновлюються мінімізацією квадратичної похибки $(y_t - Q^{test}(s_t, a_t^{test};\theta))^2$ зі стандартними механізмами буфера повторного відтворення та цільової мережі. Усереднення береться виключно на множині семантично допустимих дій $A^{test}(s')$, тому ризик-чутливе оновлення зберігає коректність переходів MAL-агрегованого графа.

ЕФЕКТИВНІСТЬ ТА ЕКСПЕРИМЕНТ

Експериментальна перевірка спрямована на оцінювання того, наскільки запропонована двоагентна модель здатна відтворювати адаптивні сценарії тестування безпеки вебзастосунків за умов часткової спостережуваності, стохастичного виявлення та обмежених ресурсів захисного агента. На відміну від задач класифікації атак, запропонований підхід використовує набори даних ToN-IoT [17] і CIC-IDS2017 [18] не як джерело готових міток для навчання класифікатора, а як основу для статистичної параметризації середовища двоагентної марковської гри.

Набір ToN-IoT використано як джерело параметризації середовища з вираженою неоднорідністю телеметрії, мережевих потоків і подій прикладного рівня. Він містить журнали IoT/IIoT-пристроїв, HTTP-взаємодії, системні події та мережевий трафік, що дозволяє моделювати веборієнтовані середовища зі змінними рівнями видимості й захисної неоднорідності. Набір CIC-IDS2017 використано для параметризації сценаріїв із реалістичним мережевим трафіком, потоковими ознаками, часовими характеристиками сесій і прикладними сценаріями атак, зокрема DoS, атаки перебором, ботнет, проникнення в мережу та вебатаки.

Початкове опрацювання даних виконувалося однаково для обох наборів. Спочатку вилучалися службові ідентифікатори, дублікати та неповні записи. Числові ознаки нормалізувалися до інтервалу [0,1], а категорійні ознаки перетворювалися на дискретні індикатори типів тестових дій. Події групувалися за часовою суміжністю, типом активності та функціональною належністю до вебкомпонента. Для ToN-IoT такими компонентами виступали HTTP-вузли, шлюзи, сервіси збору телеметрії, модулі автентифікації та механізми керування сесіями. Для CIC-IDS2017 компонентами виступали серверні обробники, мережеві потоки, вузли автентифікації, API-виклики та прикладні сесії.

Після агрегації даних формувався граф вебзастосунку $G=(N,E)$, у якому вузли N_k відповідали функціональним компонентам системи, а ребра E визначалися часовою та функціональною суміжністю подій у межах однієї сесії. Для ToN-IoT граф містив 148 агрегованих компонентів, тоді як для CIC-IDS2017 – 173 компоненти. Інтенсивність підозрілих запитів, частота невдалих автентифікацій, аномальна тривалість сесій і нетипові переходи між компонентами використовувалися для параметризації тестових атрибутів $S_{k,j}^T$. Частота блокувань, стабільність сесійних механізмів, рівень журналювання та ознаки локальної фільтрації використовувалися для формування оборонних атрибутів $S_{k,j}^D$. Параметр моніторингу $S_{k,m+1}^D$ оцінювався через локальну частоту виявлених аномалій і подій виявлення. Узагальнену схему формування середовища двоагентної гри з наборів даних показано на рисунку 2.

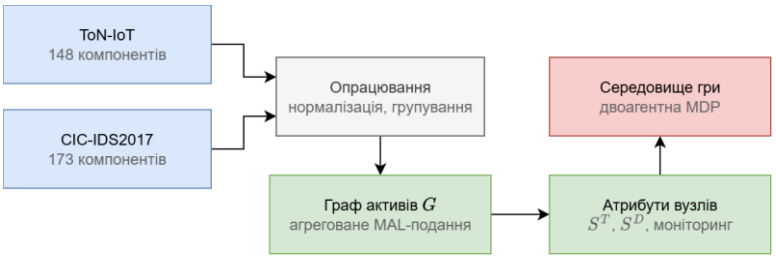

Рис.2 Схема формування середовища двоагентної гри з наборів даних

Для обох наборів даних використовувалися однакові параметри симуляційного протоколу. Кількість типів тестових дій становила $m=4$, максимальний рівень атрибутів дорівнював $w=9$, горизонт епізоду становив $T=80$, коефіцієнт дисконтування задавався як $\gamma=0.97$, а параметр ризик-чутливості дорівнював $\beta=-0.6$. Для апроксимації Q-функції використано GAT-архітектуру, запропоновану в роботі [16]. З огляду на більший простір станів двоагентної гри та потребу в стабільнішому оцінюванні ризик-чутливого критерію використано інші гіперпараметри навчання Розмір буфера повторного відтворення становив $5\cdot 10^5$ переходів,

а оновлення цільової Q-функції виконувалося кожні 2500 кроків. Для стабілізації навчання використовувалася ε -жадібна стратегія з експоненційним зменшенням параметра ε від 1.0 до 0.05. Для кожної конфігурації виконувалося 30 незалежних повторів по 1000 епізодів у кожному. Усі експерименти реалізовано мовою Python із використанням бібліотек NumPy, pandas, NetworkX і PyTorch.

Для порівняння використано три підходи. По-перше, класичний граф атак, що моделював досяжність цільового компонента без активної захисної реакції та без оновлення оборонних атрибутів. По-друге, одноагентне навчання з підкріпленням, що використовувало лише агента тестування без адаптивного захисного агента. По-третє, ризик-нейтральна частково спостережувана модель, що враховувала шум спостереження, але оптимізувала середню винагороду без TTC-орієнтованої двобічної функції винагороди. У всіх випадках використовувалися однакові графові структури, однакові стартові вузли та однаковий горизонт епізоду.

Перед порівнянням із аналогами доцільно проаналізувати часові характеристики компрометації, оскільки саме метрика TTC є центральною характеристикою запропонованої моделі. На рис. 3 для кожного набору даних подано діаграму розмаху для чотирьох моделей, що дозволяє одночасно оцінити медіани, міжквартильні інтервали, асиметрію розподілу та наявність довгих хвостів траєкторій.

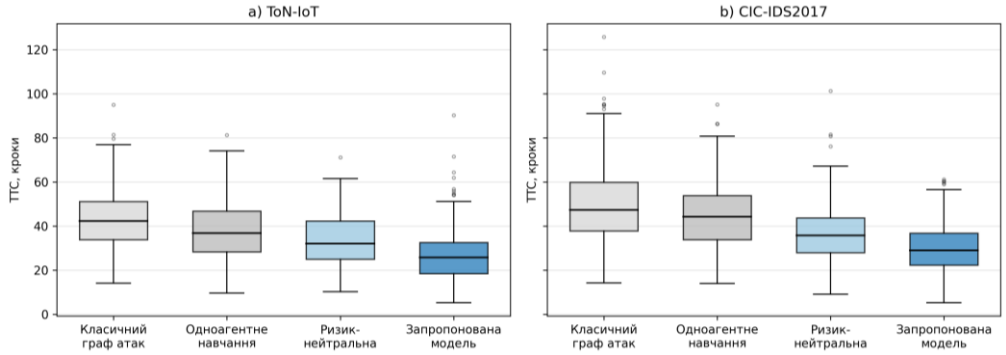

Рис.3 Розподіл часу до компрометації для порівнюваних моделей (а – для ToN-IoT, б – для CIC-IDS2017)

Як видно з рис. 3, запропонована модель систематично зміщує розподіл TTC у бік коротших критичних траєкторій. Для ToN-IoT середній TTC зменшується з 37.5±1.4 до 27.9±1.1 кроку, тоді як для CIC-IDS2017 – з 40.7±1.6 до 31.4±1.3 кроку. Найбільша різниця спостерігається в лівій частині розподілу, де зосереджені сценарії раннього досягнення вразливого стану. Це підтверджує, що ризик-чутливе навчання орієнтує політику на виявлення коротких критичних траєкторій, а не на середній сценарій взаємодії.

Окремо було досліджено вплив захисного агента на поведінку агента тестування. Для цього аналізувалася залежність між рівнем моніторингу та часткою успішних сценаріїв компрометації. У цьому експерименті структура графа та множина стартових вузлів залишалися незмінними, а змінювався лише параметр моніторингу $S^D_{k,m+1}$. На рис. 4 наведено результати для всіх порівнюваних моделей. Параметр моніторигну задає коефіцієнт, що масштабує ймовірність виявлення тестової активності на вузлі, що визначається оборонним атрибутом моніторингу $S^D_{k,m+1}$. Вищий рівень моніторингу означає більшу ймовірність виявлення дій агента тестування й, відповідно, сильніший захисний контроль.

Похибки на графіках відповідають 95 % довірчим інтервалам, отриманим за незалежними повторами симуляційного процесу.

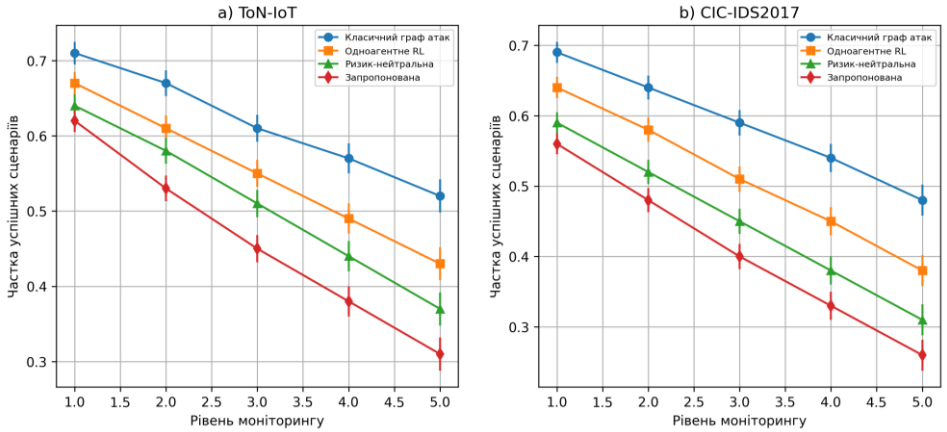

Рис.4 Вплив параметра моніторингу на частку успішних сценаріїв компрометації (а – для ToN-IoT, б – для CIC-IDS2017)

З рис. 4 видно, що збільшення рівня моніторингу призводить до зниження частки успішних сценаріїв компрометації для всіх моделей. Водночас запропонована двоагентна модель демонструє найбільшу чутливість до змін захисного профілю. Для ToN-IoT частка успішних сценаріїв зменшується приблизно з 0.62 до 0.31, тоді як для CIC-IDS2017 – з 0.56 до 0.26. Це означає, що захисний агент в межах запропонованої гри не є пасивним елементом середовища, а безпосередньо впливає на структуру переходів між станами й часові характеристики компрометації. Для класичного графа атак залежність є слабшою, оскільки така модель не враховує адаптивної реакції захисту.

Узагальнене порівняння з конкуруючими підходами наведено в таблиці 1. Розмірність визначалась як кількість унікальних агрегованих конфігурацій, що реально відвідувалися під час симуляції, а не як теоретична верхня межа.

Таблиця 1.

Порівняння запропонованої моделі з конкуруючими підходами

Модель	ToN-IoT TTC, кроки	CIC-IDS2017 TTC, кроки	Частка виявлень	Розмірність простору станів	Середній час симуляції
Класичний граф атак	44.8±1.9	48.1±2.1	0.18±0.02	$3.6\cdot10^6$	12.4±0.7 с
Одноагентне навчання з підкріпленням	37.5±1.4	40.7±1.6	0.29±0.03	$2.8\cdot10^6$	18.6±1.1 с
Частково спостережувана ризик-нейтральна модель	34.2±1.3	36.9±1.5	0.33±0.03	$2.4\cdot10^6$	25.1±1.5 с
Запропонована двоагентна модель	27.9±1.1	31.4±1.3	0.41±0.03	$2.9\cdot10^5$	7.8±0.5 с

Із таблиці 1 випливає, що запропонована модель забезпечує найнижчі значення TTC для обох наборів даних і водночас має найменший фактично відвіданий простір станів. Для ToN-IoT скорочення TTC порівняно з класичним графом атак становить 37.7%, а для CIC-IDS2017 – 34.7%. Слід зауважити, що метрика TTC обчислюється лише для епізодів, які завершилися успішною компрометацією, тоді як епізоди, перервані виявленням, відображаються у частці виявлень. Тому одночасно високі частка виявлень і низький TTC не суперечать один одному. Ризик-чутлива політика формує короткі агресивні траєкторії, що або швидко досягають вразливого стану, або завершуються виявленням захисним агентом. Слід розрізняти теоретичну потужність простору станів двоагентної гри та фактично відвіданий простір. Теоретично простір станів гри є більшим, оскільки враховує варіації як тестових, так і оборонних конфігурацій. Проте наведені в таблиці значення відображають фактично відвіданий простір – кількість унікальних станів, у яких реально побувала політика під час симуляції. Оскільки ризик-чутлива політика цілеспрямовано скорочує час до компрометації, вона відвідує лише невелику підмножину досяжних станів на коротких критичних траєкторіях, тому фактично відвіданий простір виявляється істотно меншим за теоретичну межу. Розмірність простору станів зменшується 8–12 разів порівняно з аналогами (відношення $2.4\cdot10^6$…$3.6\cdot10^6$ до $2.9\cdot10^5$ дає коефіцієнт від 8.3 до 12.4). Час симуляції скорочується меншою мірою, ніж простір станів, через додаткові витрати на оновлення захисних атрибутів, стохастичне виявлення та обчислення винагороди.

Останній експеримент спрямований на оцінювання масштабованості моделі. Для цього кількість агрегованих вебкомпонентів поступово збільшувалася від 50 до 250, а для кожного розміру графа виконувалося 30 незалежних повторів. На рис. 5 наведено залежність часу симуляції від кількості вебкомпонентів для всіх порівнюваних моделей і двох наборів даних.

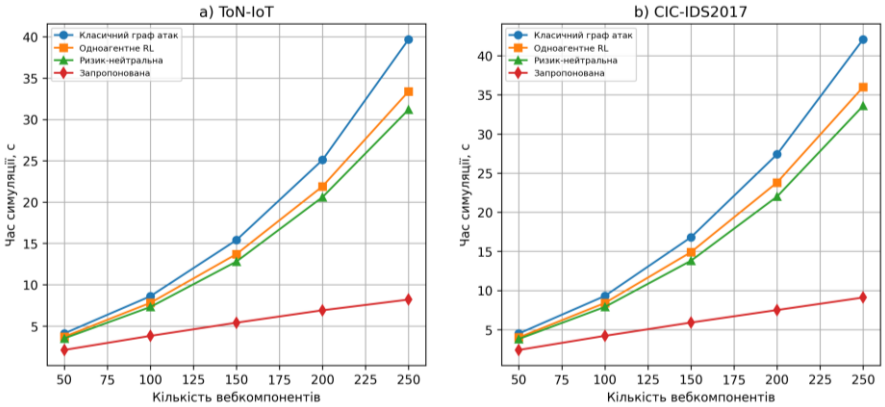

Рис.5 Залежність часу симуляції від кількості вебкомпонентів (а – для ToN-IoT, б – для CIC-IDS2017)

З рис. 5 видно, що в прийнятій реалізації класичний граф атак і одноагентне навчання з підкріпленням демонструють різке зростання часу симуляції зі збільшенням кількості вебкомпонентів. Для запропонованої двоагентної моделі час симуляції зростає повільніше й лишається в прийнятному діапазоні.

Апроксимація залежності часу симуляції від кількості компонентів на основі запропонованої моделі лінійною функцією дає коефіцієнт детермінації $R^2 = 0.94$ для ToN-IoT і $R^2 = 0.92$ для CIC-IDS2017, що свідчить про кращу масштабованість порівняно з аналогами.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

Запропоновано модель протиборства «атака–захист» у вигляді двоагентної марковської гри для тестування безпеки вебзастосунків, у якій, на відміну від односторонніх моделей атак або захисту, одночасно враховуються дії агента тестування та захисного агента в спільному просторі станів із TTC-орієнтованою двобічною функцією винагороди. Модель використовує агреговане подання системних компонентів і стохастичне задання початкових умов, що дозволило зменшити розмірність простору станів 8–12 разів і забезпечити обчислювальну керованість аналізу сценаріїв протиборства.

Результати експериментальної перевірки підтвердили ефективність запропонованого підходу для обох наборів даних. Запропонована двоагентна модель забезпечила найменший середній час до компрометації серед усіх порівнюваних підходів, а саме скорочення відносно класичного графа атак становило 37.7% для ToN-IoT і 34.7% для CIC-IDS2017, за одночасного зменшення розмірності фактично відвіданого простору. Найвища частка виявлень підтвердила, що захисний агент в межах гри є активним агентом, а не пасивним елементом середовища, а лінійна залежність часу симуляції від розміру графа засвідчила кращу масштабованість порівняно з деталізованими аналогами.

Практичне значення отриманих результатів полягає в можливості використання запропонованої моделі як формальної основи адаптивних засобів тестування безпеки вебзастосунків у середовищах із частковою спостережуваністю, стохастичним виявленням та змінними параметрами захисту. Підхід дозволяє оцінювати не лише факт досягнення вразливого стану, а й часові характеристики компрометації, що є важливим для систем раннього виявлення атак, автоматизованого тестування на проникнення та аналізу стійкості вебінфраструктури.

Обмеження роботи пов’язані з використанням агрегованого подання системних компонентів, що спрощує внутрішню структуру вебзастосунку та не враховує повний рівень деталізації міжпроцесної взаємодії, а також із симуляційним характером експериментів, у межах яких набори ToN-IoT і CIC-IDS2017 використовувалися як джерело статистичної параметризації, а не як середовище прямого виконання реальних атак.

Перспективним напрямком подальших досліджень є розробка гібридних підходів, у яких TTC-орієнтовані марковські ігри поєднуватимуться з графовими моделями вразливостей і механізмами онлайн-адаптації захисних стратегій у середовищах кіберпротиборства.

References

1. Terranova F., Lahmadi A., Chrisment I. Leveraging Deep Reinforcement Learning for Cyber-Attack Paths Prediction: Formulation, Generalization, and Evaluation. *The 27th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '24)*. ACM, 2024. P. 1–16. URL: https://doi.org/10.1145/3678890.3678902
2. Wang W., Sun D., Jiang F., Chen X., Zhu C. Research and Challenges of Reinforcement Learning in Cyber Defense Decision-Making for Intranet Security. *Algorithms*. 2022. Vol. 15, no. 4. Article 134. URL: https://doi.org/10.3390/a15040134
3. Phillips C., Swiler L. P. A graph-based system for network-vulnerability analysis. *Proceedings of the 1998 Workshop on New Security Paradigms (NSPW '98)*. ACM, 1998. P. 71–79. URL: https://doi.org/10.1145/310889.310919
4. Hu H., Liu J., Zhang Y., Liu Y., Xu X., Tan J. Attack scenario reconstruction approach using attack graph and alert data mining. *Journal of Information Security and Applications*. 2020. Vol. 54. Article 102522. URL: https://doi.org/10.1016/j.jisa.2020.102522
5. Li Q., Wu J. Optimizing the Effectiveness of Moving Target Defense in a Probabilistic Attack Graph: A Deep Reinforcement Learning Approach. *Electronics*. 2024. Vol. 13, no. 19. Article 3855. URL: https://doi.org/10.3390/electronics13193855
6. Gangupantulu R., Cody T., Rahman A., Redino C., Clark R., Park P. Crown Jewels Analysis using Reinforcement Learning with Attack Graphs. *arXiv*. 2021. URL: https://doi.org/10.48550/arXiv.2108.09358
7. Ibrahim M., Elhafiz R. Security Analysis of Cyber-Physical Systems Using Reinforcement Learning. *Sensors*. 2023. Vol. 23, no. 3. Article 1634. URL: https://doi.org/10.3390/s23031634
8. Cody T., Rahman A., Redino C., Huang L., Clark R., Kakkar A., Kushwaha D., Park P., Beling P., Bowen E. Discovering Exfiltration Paths Using Reinforcement Learning with Attack Graphs. *2022 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, 2022. P. 1–8. URL: https://doi.org/10.1109/dsc54232.2022.9888919
9. Cody T. A Layered Reference Model for Penetration Testing with Reinforcement Learning and Attack Graphs. *2022 IEEE 29th Annual Software Technology Conference (STC)*. IEEE, 2022. P. 41–50. URL: https://doi.org/10.1109/stc55697.2022.00015
10. Oh S. H., Kim J., Nah J. H., Park J. Employing Deep Reinforcement Learning to Cyber-Attack Simulation for Enhancing Cybersecurity. *Electronics*. 2024. Vol. 13, no. 3. Article 555. URL: https://doi.org/10.3390/electronics13030555
11. Zhang Y., Cai W., Chen H., Qi Z., Chen H., Liu F., He S. A Learning-Based POMDP Approach for Adaptive Cyber Defense Against Multi-Stage Attacks. *2024 IEEE International Conference on High Performance Computing and Communications (HPCC)*. IEEE, 2024. P. 1220–1227. URL: https://doi.org/10.1109/hpcc64274.2024.00164 Applebaum A., Miller D., Strom B., Korban C., Wolf R. Intelligent, automated red team emulation. Proceedings of the 32nd Annual Conference on Computer Security Applications (ACSAC '16). ACM, 2016. P. 363–373. URL: https://doi.org/10.1145/2991079.2991111
12. Jing J. Applications of Game Theory and Advanced Machine Learning Methods for Adaptive Cyberdefense Strategies. *Computational Intelligence and Neuroscience*. 2022. Vol. 2022. Article 2266171. URL: https://doi.org/10.1155/2022/2266171
13. Bitirgen K., Filik Ü. B. Markov game based on reinforcement learning solution against cyber–physical attacks in smart grid. *Expert Systems with Applications*. 2024. Vol. 255. Article 124607. URL: https://doi.org/10.1016/j.eswa.2024.124607
14. Prytula A., Kupershtein L. Modeling cyberattack scenarios as a markov decision process with a semantically constrained action space. *Cybersecurity: Education, Science, Technique*. 2026. Vol. 1, no. 33. P. 555–569. URL: https://doi.org/10.28925/2663-4023.2026.33 (in Ukrainian)..
15. Prytula A., Kupershtein L. Reward Model in Reinforcement Learning for Web Application Security Testing. Preprint. 2026. URL: https://doi.org/10.13140/RG.2.2.35201.62566 (in Ukrainian).
16. Moustafa N. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustainable Cities and Society*. 2021. Vol. 72. Article 102994. URL: https://doi.org/10.1016/j.scs.2021.102994
17. Sharafaldin I., Lashkari A. H., Ghorbani A. A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*. SciTePress, 2018. P. 108–116. URL: https://doi.org/10.5220/0006639801080116