DOI: https://doi.org/10.33216/1998-7927-2025-293-7-5-11

УДК 004.056:004.8

АГЕНТ НАВЧАННЯ З ПІДКРІПЛЕННЯМ НА ОСНОВІ SAC ДЛЯ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

Вікулов В.В.

SAC-BASED REINFORCEMENT LEARNING AGENT FOR AUTOMATED NETWORK PENETRATION TESTING

Vikulov V.V.

Загрози кібербезпеці та збитки від кіберзлочинності щороку продовжують зростати. За прогнозами, до кінця 2025 року загальні збитки досягнуть 10,5 трлн доларів, що в 3,5 рази перевищує збитки від кібербезпеки, зафіксовані в 2015 році. Зі зростанням використання штучного інтелекту зловмисниками зростає потреба в захисних інструментах, які також використовують можливості штучного інтелекту.

Ця стаття продовжує дослідницьку тенденцію застосування навчання з підкріпленням до тестування вразливостей безпеки в комп'ютерних мережах. Для досягнення цієї мети в якості тестового середовища використовується Network Attack Simulator (NASim). NASim – це симулятор, призначений для оцінки автоматизованого тестування на проникнення за допомогою підкріплювального навчання, побудований на базі фреймворку Gymnasium.

У цій статті представлено гібридний алгоритм підкріплювального навчання, який поєднує архітектуру Soft Actor-Critic (SAC) з дискретними просторами дій, що дозволяє алгоритму SAC ефективно працювати в середовищі.

Алгоритм тестувався за допомогою сценарію nasim:Small-v0. Експериментальні результати демонструють, що запропонований метод досягає декількох значних показників ефективності. По-перше, алгоритм демонструє стабільну конвергенцію протягом усього процесу навчання, що свідчить про надійну динаміку навчання. По-друге, метод демонструє виняткову ефективність у компрометації системи, вимагаючи в середньому лише 8,63 кроків під час пізніх етапів навчання для повної компрометації цільових систем. По-третє, алгоритм має ідеальний стовідсотковий рівень успішності під час фази оцінки, демонструючи надійність та стабільну продуктивність.

Крім того, алгоритм досягає середньої винагороди в розмірі 184,61 на пізніх етапах навчання, що свідчить про високу ефективність при потенційному застосуванні в галузі кібербезпеки. Однак ці результати вимагають тривалого навчання, а для більш складних сценаріїв може знадобитися ще більше часу. Це створює компроміс між обчислювальною ефективністю та якістю роботи, який необхідно враховувати при практичному впровадженні.

***Ключові слова**: навчання з підкріпленням, кібербезпека, NASim, SAC.*

Вступ. Кібербезпека стає однією з найкритичніших проблем сучасного цифрового суспільства. Кількість і складність атак щороку зростають, незважаючи на наявні інструменти захисту, що створює постійну гонку між кіберзлочинцями та фахівцями з безпеки. Ситуація ускладнюється і появою нових інструментів для атакуючої сторони, включаючи великі мовні моделі (LLM) та інші технології штучного інтелекту, які значно знижують бар'єр входу для проведення складних кібератак та дозволяють автоматизувати процеси.

Згідно зі звітом Cybersecurity Ventures [1], глобальні збитки від кіберзлочинності у 2025 році прогнозуються на рівні $10.5 трильйонів доларів США, що означає зростання більш ніж у 3.5 рази у порівнянні зі збитками 2015 року.

Застосування навчання з підкріпленням у кібербезпеці має практичний сенс через кілька ключових особливостей цього підходу. По-

перше, навчання з підкріпленням добре підходить для моделювання процесу атаки як послідовності дій, де кожен крок залежить від попередніх результатів, що точно відповідає реальній поведінці у мережі атакуючої сторони. По-друге, на відміну від класичного машинного навчання, яке потребує великих наборів даних, навчання з підкріпленням може покращуватися через взаємодію з середовищем, отримуючи сигнал винагороди лише від кінцевого результату (успішна атака чи ні). Це особливо корисно в кібербезпеці, де складно отримати якісно розмічені дані про всі етапи атак. Крім того, агенти навчання з підкріпленням можуть адаптуватися до нових сценаріїв без необхідності перенавчання на нових даних, що важливо в умовах постійної еволюції загроз.

Для практичного дослідження таких підходів необхідні спеціалізовані симуляційні середовища, які б дозволили безпечно тестувати алгоритми без ризику для реальних систем.

Network Attack Simulator (NASim) [2] реалізовано як середовище Gymnasium, що забезпечує стандартизований інтерфейс для алгоритмів навчання з підкріпленням. Хоча NASim є спрощеною абстракцією реальних мережевих середовищ, він надає достатню складність для навчання агентів виконанню багатоетапних атак. Використання фреймворку Gymnasium дозволяє легко інтегрувати різні алгоритми підкріплювального навчання та забезпечує консистентний API для взаємодії агента з середовищем, що критично важливо для порівняння ефективності різних підходів до автоматизації кібератак.

Метою роботи є дослідження ефективності сучасних алгоритмів навчання з підкріпленням, зокрема адаптованого Soft Actor-Critic (SAC) для автоматизації мережевих атак у симуляційному середовищі NASim. Основні завдання включають адаптацію SAC для роботи з дискретним простором дій NASim, порівняльний аналіз ефективності різних підходів та оцінку їх потенціалу для практичного застосування в кібербезпеці.

Огляд літератури. Schwartz та Kurniawati (2019) [2] вперше запропонували платформу NASim для навчання з підкріпленням у тестуванні на проникнення. Вони використали Q-навчання і показали, що воно може знаходити оптимальні шляхи атак у різних мережах. Але метод добре працював тільки для маленьких мереж з невеликою кількістю дій.

Becker та ін. (2024) [3] порівняли A3C, DQN і Q-learning у NASim. A3C зміг вирішити всі сценарії NASim та узагальнювати знання між різними варіантами середовища. DQN і Q-learning показали гірші результати, особливо коли тренувалися на кількох сценаріях одночасно.

Li, Zhang та Yang (2024) [4] розробили EPPTA – метод на основі SAC з модулем “belief state” для роботи в умовах неповної інформації. Він навчався у 20 разів швидше за інші та отримував майже максимальну винагороду у сценаріях NASim, у тому числі в “Small” сценарії.

Tran та ін. (2022) [5] представили CRLA – каскад агентів, які розбивають простір дій на менші підзадачі. Хоча вони тестували це у CybORG, підхід добре підходить і для NASim-подібних завдань.

Janisch, Pevný та Lisý (2023) [6] створили NASimEmu, щоб вирішити головну проблему NASim: агенти добре працюють у симуляції, але погано у реальних мережах. NASimEmu дозволяє навчати агента у швидкій симуляції, а потім тестувати його на реальних віртуальних машинах. Експерименти показали, що агенти успішно переносяться з симуляції у реальне середовище.

CyberBattleSim [7] є експериментальним симуляційним середовищем, розробленим Microsoft Research для дослідження застосування навчання з підкріпленням у кібербезпеці [4]. На відміну від NASim, який фокусується на абстрактних представленнях мережевих топологій, CyberBattleSim моделює більш реалістичні сценарії кібератак з детальним відтворенням процесів lateral movement та privilege escalation. Середовище побудоване на основі фреймворку OpenAI Gym. Ключовою особливістю CyberBattleSim є можливість моделювання складних багатоетапних атак з урахуванням різних типів вразливостей, включаючи експлойти нульового дня та соціальну інженерію. Однак складність середовища призводить до значно більших вимог до обчислювальних ресурсів та часу навчання порівняно з більш абстрактними симуляторами типу NASim. Це робить CyberBattleSim особливо цінним для дослідження складних сценаріїв, але менш практичним для швидкого прототипування та тестування нових алгоритмів.

Wang та ін. (2023) [8] запропонували DQfD-AIPT, який тестувався на розроблених ними сценаріях для CyberBattleSim. У сценаріях з “honeypot”-вузлами він знаходив ціль швидше

та уникав пасток, отримуючи вищу винагороду, ніж звичайний DQN.

Методологія. Network Attack Simulator має досить низькі нагороди, що робить застосування навчання з підкріпленням особливо складним завданням. Агент отримує позитивний сигнал лише після успішного виконання повного сценарію атаки, при цьому NASim навіть не винагороджує за проміжні кроки, необхідні для компрометування окремих систем. Це означає, що агенту необхідно виконати певну послідовність правильних дій без отримання підкріплюючих сигналів на проміжних етапах, при цьому більшість випадкових дій не призводять до прогресу. Така структура винагород створює проблему розрідженої винагороди, що є реалістичним відображенням реальних задач кібербезпеки, де успіх досягається лише через ретельне планування та точне виконання багатоетапних атак.

У запропонованому рішенні не було застосовано додаткових винагород, тобто тільки ті, що були передбачені розробником NASim. Також це необхідно для порівняння з результатами інших дослідників.

Сценарій nasim:Small-v0 було обрано для дослідження з декількох ключових причин.

По-перше, сценарій відтворює типову мережу малих та середніх розмірів, яка на сьогоднішній день може відповідати конфігурації мереж досить великих компаній та корпорацій. Це робить його особливо актуальним для моделювання реальних кібератак та оцінки ефективності захисних механізмів.

По-друге, складність та розмір сценарію є оптимальними для навчання моделей машинного навчання, оскільки він містить достатню кількість вузлів для створення складних шляхів атак, не є надмірно складним, що дозволяє ефективно навчати модель, та забезпечує певний баланс між реалістичністю та обчислювальною складністю.

Обраний сценарій складається з 8 комп'ютерів, розподілених між двома основними операційними системами: Windows та Linux.

На емульованих комп’ютерах запущено 3 сервіси: FTP, SSH та HTTP. Ці сервіси представляють типові вектори атак: FTP – через слабкі паролі та неправильні конфігурації, SSH – для горизонтального переміщення в мережі, HTTP – як початкову точку атак через веб-інтерфейси.

Серед процесів, що виконуються в емульованому середовищі, використано Apache Tomcat Server та DACLSVC. Tomcat є сервером Java-додатків, а DACLSVC – це Windows-сервіс для управління доступом.

Запропонований алгоритм також було протестовано для сценарію Nasim:Medium-v0, який складається з мережі у 2 рази більшої за розміром (16 комп'ютерів). Проте навчання на більшому сценарії потребувало значно більше обчислювальних ресурсів та часу для досягнення аналогічного рівня конвергенції. Це зумовлено експоненційним зростанням простору станів та дій при збільшенні розміру мережі. Враховуючи обмежені обчислювальні можливості та необхідність проведення множинних експериментів для валідації результатів, було прийнято рішення зосередитися на сценарії nasim:Small-v0 як оптимальному компромісі між складністю задачі та практичністю експериментування.

У даному дослідженні було обрано алгоритм Soft Actor-Critic (SAC) з адаптацією для дискретного простору дій. Цей вибір обумовлений кількома ключовими факторами, що роблять SAC особливо придатним для задач кібербезпеки.

По-перше, NASim використовує дискретний простір дій, де агент обирає конкретну дію/атаку з фіксованого набору можливих дій. Оригінальний SAC розроблений для роботи з неперервними діями, тому виникла необхідність адаптації алгоритму. Для вирішення цієї проблеми було реалізовано DiscreteToBoxActionWrapper, який перетворює дискретний простір дій у неперервний простір Box(0,1) розмірністю n, де n відповідає кількості можливих дій. Агент генерує неперервний вектор, після чого обирається дія з найвищою ймовірністю.

По-друге, ключовою перевагою SAC є вбудована entropy regularization, яка заохочує агента до дослідження різноманітних стратегій атак.

По-третє, для реалізації було обрано SAC з бібліотеки stable-baselines3 (SB3), яка забезпечує гнучке налаштування основних гіперпараметрів. Це дозволяє легко експериментувати з різними конфігураціями навчання, включаючи швидкість навчання, розміри буферів та архітектуру нейронних мереж, без необхідності модифікації основного коду. Така реалізація особливо важлива для швидшого тестування різних налаштувань при виконанні дослідницьких завдань.

Таблиця 1

Гіперпараметри алгоритму SAC (для SB3)

Параметр	Значення за замовчуванням	Опис
Learning Rate	3×10^{-4}	Швидкість навчання
Discount Factor (γ)	0.99	Коефіцієнт дисконтування
Replay Buffer Size	1,000,000	Розмір буфера досвіду
Batch Size	256	Розмір батча
Soft Update Coefficient (τ)	0.005	Коефіцієнт м'якого оновлення target мереж
Hidden Layer Size	256	Кількість нейронів у прихованих шарах
Entropy Coefficient	auto	Автоматичне налаштування коефіцієнта ентропії

Навчання агента проводилося протягом 509,000 кроків взаємодії з середовищем. Цей обсяг було обрано як компроміс між достатністю для конвергенції та обчислювальними обмеженнями. Хоча була закладена можливість використання CUDA, обчислення виконувалися швидше на CPU через кілька факторів: відносно невеликий розмір нейронних мереж (два шари по 256 нейронів, що забезпечує достатню репрезентативну потужність для навчання складних стратегій атак без надмірного ускладнення моделі), помірний розмір батча (256) та overhead від постійної передачі даних між CPU та GPU.

Після завершення навчання ефективність агента оцінюється на 100 незалежних епізодах з детерміністичною політикою (без exploration). Система винагород у NASim побудована таким чином, що агент отримує позитивну винагороду за успішне досягнення цільового стану (компрометація цільової системи) та негативну або нульову винагороду за невдалі спроби або проміжні кроки. Таблиця 2 описує використані метрики оцінювання.

Таблиця 2

Метрики оцінювання ефективності

Метрика	Спосіб обчислення
Рівень успішності – відсоток успішних атак	(Кількість успішних епізодів / 100) × 100%
Середня кількість кроків за епізод	Середнє арифметичне кроків по всіх епізодах
Винагорода – сумарна винагорода за епізод	Сума всіх винагород протягом епізоду

Успішність епізоду визначається через виклик методу env.unwrapped.goal_reached(), який повертає True, якщо агент досяг цільового стану – повного злому системи. Average Steps дозволяє оцінити ефективність стратегії: менша кількість кроків для досягнення мети свідчить про більш оптимальну поведінку агента.

Після завершення навчання натренована модель зберігається для подальшого аналізу та використання. Підтримується логування метрик у TensorBoard для візуального аналізу процесу навчання.

Виклад основного матеріалу дослідження. Для валідації запропонованого підходу було проведено декілька експериментів із тренування SAC-агента на сценарії nasim:Small-v0. Кожен експеримент підтверджувався декількома запусками. Обидва експерименти виконувалися з однаковими гіперпараметрами, окрім learning rate та кількості кроків.

Експеримент 1 виконувався на понад 500 тисяч кроків з learning rate $3*10^{-4}$.

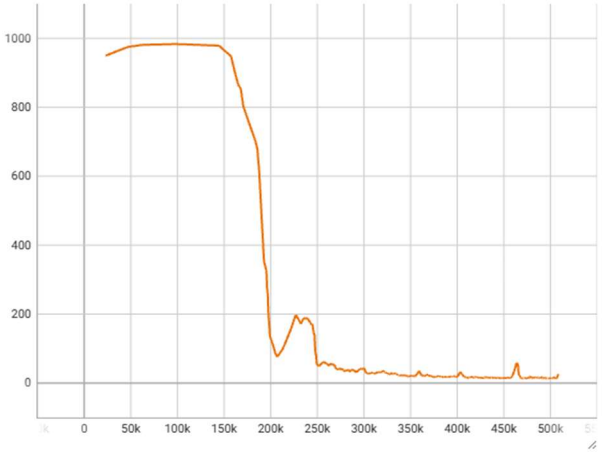

Рис. 1. Графік кількості кроків, необхідних для повного компрометування мережі. Експеримент 1

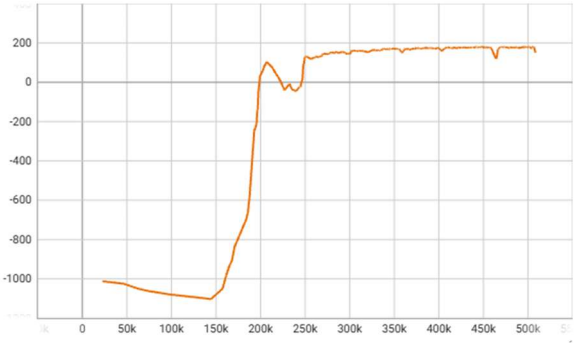

Рис. 2. Графік отриманих нагород під час навчання. Експеримент 1

Експеримент 2 виконувався на понад 1.4 мільйони кроків з learning rate $5*10^{-4}$. Цікаво, що після мільйона кроків все ще продовжувалися покращуватися метрики кількості кроків та винагороди, проте швидкість цього покращення була незначною.

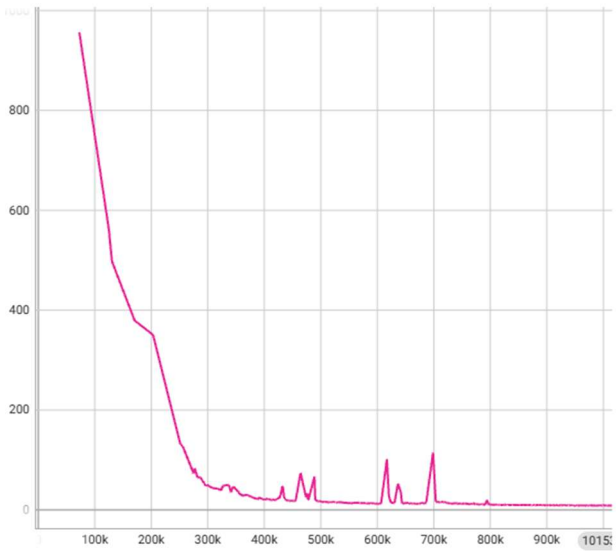

Рис. 3. Графік кількості кроків, необхідних для повного компрометування мережі. Експеримент 2

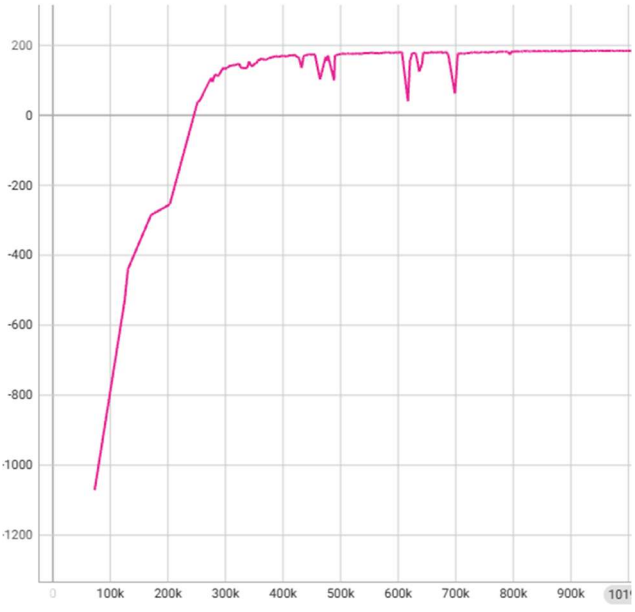

Рис. 4. Графік отриманих нагород під час навчання. Експеримент 2

Особливо цінним є той факт, що запропонований підхід досягає 100% рівня успішності в обох експериментах на фінальних етапах навчання. Це означає, що в контрольованому середовищі агент здатен гарантовано виконувати поставлені завдання, що є критично важливим для автоматизованих систем безпеки.

Таблиця 3

Порівняльні результати різних експериментів SAC

Метрика	SAC експеримент 1	SAC експеримент 2	Оптимальне значення
Середня кількість кроків (останні 100 епізодів)	14.68	**8.63**	8.0
Мінімальна кількість кроків	13.11	**8.40**	8.0
Середня винагорода (останні 100 епізодів)	177.14	**184.61**	186.0
Максимальна винагорода	179.34	**185.09**	186.0

Оптимальне значення вказано в конфігурації сценарію [9]. Результати показують стабільну збіжність алгоритму до оптимального рішення. Як випливає з таблиці 3, найкращий експеримент (другий) досяг середньої кількості кроків 8.63 на останніх 100 епізодах, що становить відхилення лише 7.9% від теоретичного оптимуму (8 кроків). Середня винагорода 184.61 балів становить 99.2% від максимально можливої винагороди (186 балів), що свідчить про здатність агента знаходити майже оптимальні шляхи атаки.

Порівняємо у таблиці 4 досягнуті результати з результатами інших дослідників, наведеними у статті [10]:

Таблиця 4

Порівняльний аналіз алгоритмів для вирішення сценарію nasim:Small-v0

Алгоритм	Середня нагорода	Середня кількість кроків
SAC (дане дослідження, експеримент 2)	184.61	8.63
EPPTA	184.9	8.4
Recurrent PPO	184.8	8.4
NDSPI-DQN	174.0	17.3
HA-DQN	-256.2	464.9

Наведені результати підтверджують ефективність запропонованої адаптації SAC для дискретних просторів дій у контексті кібербезпеки.

Як випливає з рисунків 1-4, процес навчання характеризується:

1. Відносно швидкою початковою збіжністю: алгоритм демонструє стрімке покращення ефективності на досить ранніх етапах навчання.

2. Стабільною фінальною продуктивністю: останні епізоди показують консистентні результати без значних коливань.

Водночас, важливо визнати обмеження поточного дослідження. Тестування проводилося виключно на сценарії nasim:Small-v0, який, хоча і репрезентативний для багатьох реальних ситуацій, все ж таки є спрощеною абстракцією складних корпоративних мереж. Масштабованість підходу на більші та складніші сценарії залишається відкритим питанням, яке потребує подальшого дослідження.

Висновки. У даному дослідженні було успішно адаптовано алгоритм Soft Actor-Critic (SAC) для роботи з дискретним простором дій у середовищі Network Attack Simulator (NASim) та продемонстровано його високу ефективність для автоматизації мережевих атак.

Запропонована адаптація SAC показала відмінні результати на сценарії nasim:Small-v0. Алгоритм досяг середньої кількості кроків 8.63 при оптимальному значенні 8.0 (відхилення лише 7.9%) та середньої винагороди 184.61 балів при максимально можливій винагороді в 186 балів (99.2% від оптимуму). Важливим досягненням є 100% рівень успішності на фінальних етапах навчання в обох експериментах, що демонструє надійність запропонованого підходу.

Порівняльний аналіз з наявними рішеннями показав, що адаптований SAC досягає результатів, схожих з алгоритмами EPPTA та Recurrent PPO, значно перевершуючи базові підходи типу PPO та DQN.

Результати дослідження підтверджують високий потенціал застосування алгоритмів навчання з підкріпленням для вирішення практичних задач автоматизованого тестування на проникнення. Запропонована адаптація SAC забезпечує стабільні та близькі до оптимальних результати, що робить її перспективною основою для розробки більш досконалих систем кібербезпеки та автоматизованого виявлення вразливостей у мережевій інфраструктурі.

Література

1. Cybersecurity Ventures. *Cyberwarfare 2021 Report*. 2021. URL: https://cybersecurityventures.com/wp-content/uploads/2021/01/Cyberwarfare-2021-Report.pdf (дата звернення: 13.08.2025).
2. Qu, S., Du, W., Chen, C., Li, B., & Qiu, M. *A survey on reinforcement learning applications in cybersecurity*. arXiv preprint arXiv:1905.05965, 2019. URL: https://arxiv.org/abs/1905.05965 (дата звернення: 13.08.2025).
3. Becker, N., Reti, D., Ntagiou, E. V., Wallum, M., & Schotten, H. D. *Evaluation of Reinforcement Learning for Autonomous Penetration Testing using A3C, Q-learning and DQN*. arXiv, 2024. doi: 10.48550/arXiv.2407.15656.
4. Li, Z., Zhang, Q., & Yang, G. *EPPTA: Efficient partially observable reinforcement learning agent for penetration testing applications*. *Engineering Reports*, 2024. doi: 10.1002/eng2.12818.
5. Tran, K., Standen, M., Kim, J., Bowman, D., Richer, T., Akella, A., & Lin, C. T. *Cascaded reinforcement learning agents for large action spaces in autonomous penetration testing*. *Applied Sciences*, 2022, 12(21), 11265. doi: 10.3390/app122111265.
6. Janisch, J., Pevný, T., & Lisý, V. *NASimEmu: Network attack simulator & emulator for training agents generalizing to novel scenarios*. arXiv preprint arXiv:2305.17246, 2023. URL: https://arxiv.org/abs/2305.17246 (дата звернення: 13.08.2025).
7. Microsoft. *CyberBattleSim: An experimentation research platform to investigate automated agents operating in simulated enterprise environments*. 2021. URL: https://github.com/microsoft/CyberBattleSim (дата звернення: 13.08.2025).
8. Wang, Y., Li, Y., Xiong, X., Zhang, J., Yao, Q., & Shen, C. *DQfD-AIPT: An intelligent penetration testing framework incorporating expert demonstration data*. *Security and Communication Networks*, 2023, 5834434. doi: 10.1155/2023/5834434.
9. Schwartz, J. *Network Attack Simulator — small.yaml scenario*. GitHub, 2023. URL: https://github.com/Jjschwartz/NetworkAttackSimulator/blob/4f26de37cfdc3e4553ed8b7484c4db8e2924bdea/nasim/scenarios/benchmark/small.yaml (дата звернення: 13.08.2025).
10. Li, Z., Zhang, Q., & Yang, G. *EPPTA: Efficient partially observable reinforcement learning agent for penetration testing applications*. *Engineering Reports*, 2025, 7(1): e12818. doi: 10.1002/eng2.12818.

Vikulov V.V. SAC-based reinforcement learning agent for automated network penetration testing

Cybersecurity threats and cybercrime damage continue to escalate annually. Projections indicate that total damages will reach $10.5 trillion by the end of 2025, representing a 3.5-fold increase from the cybersecurity damages recorded in 2015. With the rise of AI adoption by malicious actors, there is an increasing need for defensive tools that also leverage AI capabilities.

This paper continues the research trend of applying reinforcement learning to penetration testing for identifying security vulnerabilities in computer networks. To achieve this objective, the Network Attack Simulator (NASim) is employed as a testing environment. NASim is a simulator designed for evaluating automated penetration testing through reinforcement learning, built on the Gymnasium framework.

This paper presents a hybrid reinforcement learning algorithm specifically, which combines the Soft Actor-Critic (SAC) architecture with discrete action spaces, thereby enabling the SAC algorithm to operate effectively within environment.

The algorithm underwent evaluation using the nasim:Small-v0 scenario. Experimental results demonstrate that the proposed method achieves several noteworthy performance metrics. First, the algorithm exhibits stable convergence behavior throughout the training process, indicating robust learning dynamics. Second, the method demonstrates exceptional efficiency in system compromise, requiring an average of 8.63 steps during late training episodes to fully compromise target systems. Third, the algorithm maintains a perfect 100% success rate during the evaluation phase, demonstrating reliable and consistent performance.

Additionally, the algorithm achieves an average reward of 184.61 in later training stages, indicating strong performance for potential cybersecurity applications. However, these results require extensive training time, with potentially even longer training periods needed for more complex scenarios. This creates a trade-off between computational efficiency and performance quality that must be considered for practical implementation.

Keywords: *reinforcement learning, cybersecurity, NASim, SAC.*

Вікулов Владислав Вікторович – аспірант, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», v.vikulov@kpi.ua

Стаття подана 05.08.2025.