КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (23), 2024
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2024.23.1730
УДК 004.8

Толкачова Анастасія Юріївна
студент спеціальності «Кібербезпека»
Національний Університет «Львівська Політехніка», Львів, Україна
ORCID 0000-0002-8196-7963
anastasiia.tolkachova.mkbst.2022@lpnu.ua

Посувайло Максим-Микола Володимирович
студент спеціальності «Кібербезпека»
Національний Університет «Львівська Політехніка», Львів, Україна
ORCID 0009-0000-8553-2142
posuvaylom@gmail.com

ТЕСТУВАННЯ НА ПРОНИКНЕННЯ З ВИКОРИСТАННЯМ ГЛИБОКОГО НАВЧАННЯ З ПІДКРІПЛЕННЯМ

Анотація. Традиційно, тестування на проникнення виконується експертами, які вручну моделюють атаки на комп'ютерні мережі з метою оцінки їх захищеності та виявлення вразливостей. Проте, сучасні дослідження підкреслюють значний потенціал автоматизації цього процесу через глибоке навчання з підкріпленням. Розробка автоматизованих систем тестування обіцяє значне підвищення точності, швидкості та ефективності виявлення та усунення вразливостей. У фазі підготовки до тестування, можливе використання штучного інтелекту для автоматичного створення реалістичної топології мережі, включаючи розробку дерева можливих атак. Застосування методів глибокого навчання, як-от Deep Q-Learning, дозволяє системі визначати оптимальні шляхи атаки, що робить процес вторгнення більш стратегічним та обґрунтованим. Автоматизовані системи тестування на проникнення можуть виступати як ефективні навчальні інструменти для підготовки спеціалістів у галузі кібербезпеки. Вони дозволяють імітувати атаки в контрольованому навчальному середовищі, пропонуючи користувачам аналізувати різні стратегії та методики вторгнення, а також слугують засобами для тренування виявлення та реагування на реальні атаки. Такий підхід сприяє глибокому розумінню потенційних загроз і розвиває навички ефективної оборони від них. Крім того, використання машинного навчання може допомогти у вирішенні проблеми великої кількості помилкових позитивних результатів, що є загальнопоширеною проблемою в традиційних системах безпеки. Глибоке навчання з підкріпленням надає можливість для створення більш адаптивних систем тестування, які здатні самонавчатися та адаптуватися до змінних моделей загроз. Такі системи стають не лише більш ефективними, але й здатні працювати з меншою кількістю помилок, зменшуючи навантаження на людський фактор. Завдяки цьому, вони можуть виявити непомічені людиною вразливості, забезпечуючи більш глибокий та всебічний аналіз безпеки. Такий підхід має потенціал революціонізувати галузь кібербезпеки, пропонуючи нові стратегії захисту інформаційних систем та створення більш надійних мережевих структур.

Ключові слова: тестування на проникнення; штучний інтелект; машинне навчання; навчання з підкріпленням; аудит мережевої безпеки; наступальна кібербезпека; оцінка вразливостей.

ВСТУП

З огляду на все більше поширення комп'ютерних мереж та зростаючу кількість випадків порушення безпеки, кібербезпека стала надзвичайно важливою. Проактивний підхід є найбільш ефективним способом боротьби з цими проблемами, і одним із засобів такого підходу є тестування на проникнення. Цей метод оцінює безпеку системи та

допомагає уникнути можливих випадків порушення безпеки. Тестування на проникнення є одним з найважливіших методів, який застосовують організації для підвищення рівня безпеки і захисту від кіберзагроз. Крім того, тестування є важливою частиною законодавства та ряду положень, наприклад, підпадаючих в певні області законодавства, таких як Health Insurance Portability and Accountability Act (HIPAA) та Payment Card Industry Data Security Standard (PCI DSS) [4], [10]. Урядові агенції також можуть вимагати пентестингу для інформаційних систем, які обробляють конфіденційні дані.

Тестування на проникнення вимагає значного часу та навчання для ефективного та якісного виконання, існує значний дефіцит кваліфікованих фахівців з кібербезпеки. Застосування методів штучного інтелекту в галузі кібербезпеки є одним із шляхів розв'язання цієї проблеми, зокрема методів машинного навчання для автоматизації тестування на проникнення. Чинні підходи до автоматизованого тестування на проникнення базуються на методах, яким потрібні моделі результатів експлуатації. Швидка зміна ландшафту кібербезпеки та різноманітність архітектур, роблять створення і підтримку актуальних моделей викликом. Тому стаття досліджує застосування методів машинного навчання для розробки автоматизованої системи тестування на проникнення.

Постановка проблеми. Основна проблема, з якою стикається RL, полягає в тому, що воно вимагає багато взаємодій з навколишнім середовищем для того, щоб навчитися оптимальну політику. Ця особливість призвела до того, що багато успішних застосувань RL було зроблено в імітаційних або ігрових середовищах, де агенти RL здатні швидко взаємодіяти з навколишнім середовищем. Наразі не існує вільно доступного імітаційного середовища для тестування проникнення, яке можна було б використовувати для навчання та тестування агентів RL. Бажання застосувати RL для автоматизованого тестування, а також відсутність середовища для тренувань зумовили розробку та мету цього дослідження.

Аналіз останніх досліджень і публікацій. Оскільки сфера автоматизації та вдосконалення тестування на проникнення знаходиться на межі між кібербезпекою та дослідженнями в галузі штучного інтелекту, було розглянуто кілька напрямків досліджень, які розвивалися в різних дослідницьких галузях та методологіях автоматизованого планування (підгалузь штучного інтелекту). Ранні дослідження були зосереджені на моделюванні проникнення у вигляді графів атак і дерев рішень, що відображають погляд на практику тестування як на послідовне прийняття рішень. На практиці, більшість робіт були більше пов'язані з оцінкою вразливостей, ніж з тестуванням [5].

Метою статті є покращення процесу виконання пошуку вразливостей із застосуванням машинного навчання для дослідження внутрішньої мережі.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

У залежності від конкретної методології тестування на проникнення, може бути визначено кілька етапів для проведення пентесту. Зміст кожної фази може залежати від специфіки цільової системи та мети тестування.

Початкові етапи тестування зазвичай пов'язуються з збором інформації про цільову систему, оскільки це допомагає тестувальникам отримати краще розуміння структури та потенційних вразливостей цільової системи. На цій стадії можлива техніка сканування мережі для виявлення активних пристроїв, перебір відкритих портів та протоколів, інструментальне зібрання інформації, та інші техніки.

Після збору інформації, тестувальник активно атакує цільову систему за допомогою різних підходів для виявлення можливих вразливостей. Цей етап може обмежуватися простим перебором слабких паролів або використовувати більш технічні методи, такі як аналіз переповнень буфера, перехоплення пакетів та інші техніки.

Після успішної атаки і отримання доступу до цільової системи, тестувальник отримує можливість зібрати додаткову інформацію, щоб підвищити рівень контролю над системою та знайти можливості для подальшої експлуатації. На цій стадії можливе використання інструментарію для збору конфіденційної інформації, навігації по файловій системі та реєстру, та запуску додаткових інструментів для отримання додаткової інформації.

У деяких випадках підвищення привілегій може мати різні етапи або бути об'єднаним з іншими етапами. На цьому кроці можливе використання технік, таких як введення в систему власного командного рядка, використання експлойтів для отримання вищих привілегій, та інші методи.

Останній етап пов'язаний з рухом по мережі та може включати сканування додаткових мереж та систем в організації для виявлення можливих вразливостей. На цій стадії можливе використання технік, таких як розгортання торгових автоматів та інших атак на підсистеми мережі.

Всі етапи тестування на проникнення пов'язані між собою, тому для досягнення запланованої мети необхідно взаємодіяти з кожною фазою [11]. Краща реалізація цих етапів допомагає зміцнити безпеку системи та запобігти можливим негативним наслідкам.

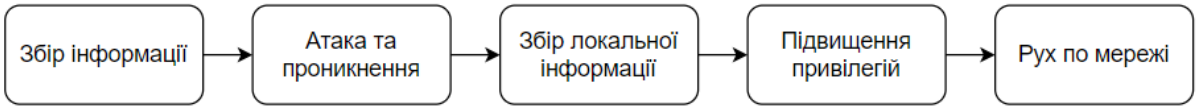

Рис. 1. Фази проведення тестування на проникнення

Інструменти для тестування на проникнення. Далі буде описано інструменти, алгоритми та методи, які ми використовуватимемо як компоненти нашого фреймворку автоматизованого тестування на проникнення.

1. Shodan — це пошуковий інструмент, який дозволяє виявляти пристрої, підключені до Інтернету, та інфраструктуру, таку як маршрутизатори, сервери та IoT-пристрої [13]. Він використовує спеціалізований сканер, який сканує Інтернет та збирає дані про відкриті порти, сервіси та протоколи, які використовуються на виявлених пристроях. Shodan може використовуватися для пошуку небезпечних пристроїв, проведення дослідження ринку, виявлення конфіденційної інформації та інших завдань. Цей інструмент зазвичай використовують пентестери та зловмисники для збору даних про потенційні мішені. За даними CNN Money, база даних Shodan містить інформацію про більше ніж 500 мільйонів мережевих пристроїв, таких як їх IP-адреси та список запущених сервісів [3]. В нашому дослідженні ми використовуємо Shodan для збору даних, необхідних для створення реалістичних мережевих топологій, які використовуються в нашому алгоритмі.
2. Модель атакуючого дерева є структурним методом аналізу IT-загроз та вразливостей, який ґрунтується на систематичному зображенні можливих атак на певну ціль та їхніх способів, використовуючи деревоподібну

структуру. Головна мета, тобто об'єкт аналізу, знаходиться у центрі атакуючої моделі, і від неї розгалужуються гілки дерева до підцілей, які є досяжними під час атаки. Дерево потім рекурсивно розгалужується вглиб, демонструючи потенційні перемикання, значення та можливості атаки. Це забезпечує формальну методологію для моделювання та аналізу різних рівнів матеріальних засобів, які може використовуватися для виявлення та захисту від кібератак. Модель атакуючого дерева є також корисним інструментом для навчання та просвітництва в області кібербезпеки, сприяючи кращому розумінню принципів виявлення та захисту від кібератак серед керівників, фахівців з IT-безпеки та кінцевих користувачів. У такий спосіб модель атакуючого дерева є важливим внеском в розробку обговорень, пов’язаних з цією галуззю науки, і вона буде продовжувати безпосередньо впливати на формування політики та захисту від кібератак [1].

3. MulVAL — це інструмент, який дозволяє фахівцям з кібербезпеки оцінювати різні типи атак та розуміти, як взаємодія між вразливостями може привести до порушення безпеки мережі. Інструмент використовує логічне представлення взаємозв’язків між вразливостями, політиками безпеки та конфігурацій системи, що дозволяє проводити ефективні та структуровані аналізи впливу вразливостей на безпеку системи. MulVAL дозволяє знаходити всі можливі шляхи для заданої топології мережі та має відкритий вихідний код для генерації дерева атак [12].

За допомогою комбінації інструментів автоматизації тестування та моделей аналізу вразливостей можна підвищити ефективність процесу забезпечення безпеки інформації. Проте, коли ми знаходимо вразливий пристрій, що доступний через Інтернет, алгоритми машинного навчання можуть допомогти найефективніше його експлуатувати. Класифікація алгоритмів машинного навчання залежить від характеру використовуваних даних для побудови моделі (навчальні дані) і може включати контрольоване машинне навчання, неконтрольоване машинне навчання, напівконтрольоване машинне навчання та машинне навчання з підкріпленням.

Навчання з підкріпленням є методом машинного навчання, в якому розробники створюють систему винагород за бажану поведінку та покарання за негативну. Цей метод включає присвоєння позитивних значень бажаним діям з метою стимулювання агента їх використовувати, а також присвоєння негативних значень для небажаної поведінки з метою її уникнення. Під таким методом програмується агент для пошуку довгострокових та максимальних винагород з метою досягнення оптимального рішення [8].

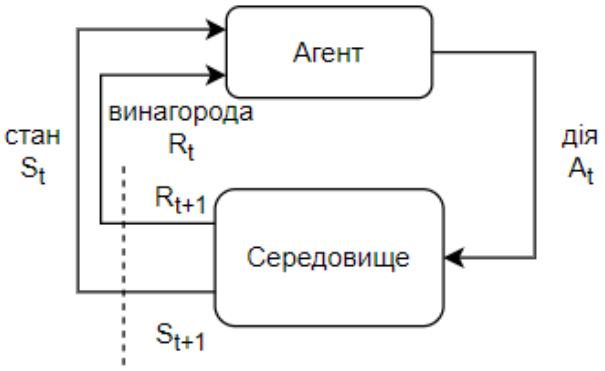

Рис. 2. Схема роботи навчання з підкріпленням

ISSN 2663 - 4023

Основні компоненти системи навчання з підкріпленням включають такі елементи:

1. Агент — це елемент, який взаємодіє з середовищем та виконує дії. Агент може бути як вчителем, так і приймачем рішень;
2. Середовище — це простір, в якому агент рухається і навчається. Воно реагує на дії агента та надає йому нові ситуації;
3. Стан (S_t) — це поточна ситуація, яку середовище повертає агенту. Кожен стан визначається в конкретний момент часу t;
4. Винагорода (R_t) — це число, яке агент отримує як результат виконання дії. Винагорода може бути позитивною або негативною;
5. Дія (A_t) — це те, що агент вирішує зробити в конкретному стані. Виконання дії впливає на середовище, яке переводить агента в новий стан і надає винагороду чи штраф.

Передбачається, що агент розпочинає взаємодію з середовищем, отримуючи стан S_t і вибираючи дію A_t, яку потрібно виконати. Після цього середовище повертає новий стан S_{t+1} та винагороду R_{t+1}, які допомагають агентові навчитися вибирати кращі дії в майбутньому. Deep Reinforcement Learning — підгалузь штучного інтелекту, яка поєднує техніки глибокого навчання та навчання з підкріпленням. Цей підхід дає змогу навчити агентів виконувати складні завдання у різноманітних динамічних середовищах, навчаючись на основі власного досвіду.

Підкріплене навчання включає такі елементи, як агент, середовище, дії, стани та винагороди. Головна мета агента полягає в максимізації загальної очікуваної винагороди протягом періоду, коли він взаємодіє зі своїм середовищем. Цей процес навчання заснований на послідовному виборі дій, які приводять до формування тактики або стратегії, що керує поведінкою агента в середовищі на основі раніше обраного стану.

Згодом буде описаний фреймворк для проведення автоматизованих тестів на проникнення, створений та запроваджений за допомогою методів глибокого навчання з підкріпленням. Фреймворк містить три основні компоненти:

- Навчальні дані: ця частина створює необхідні дані для алгоритму DQN, які служать вхідними даними для навчання;
- Модуль DQN: ця частина навчає алгоритм DQN та використовує навчену модель для проведення тестів на проникнення;
- Інструменти для проникнення: ця частина є обгорткою для зовнішніх інструментів, які використовуються для виконання дій на реальних системах.

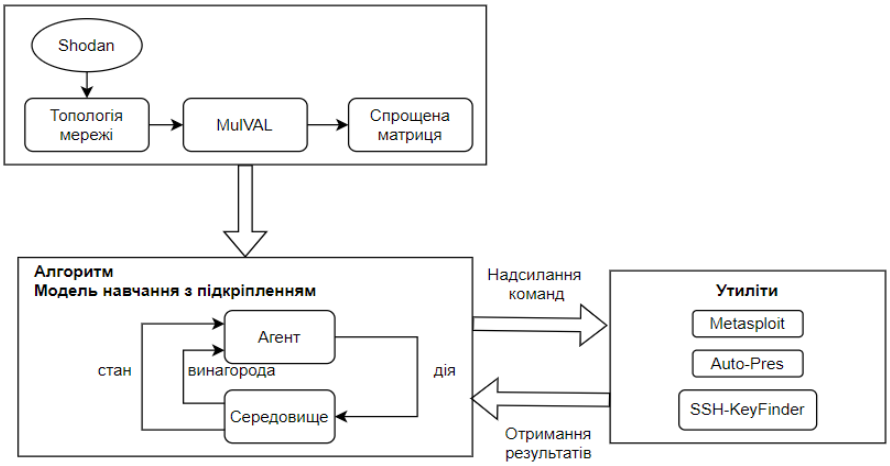

Рис. 3. Архітектура автоматизованого пошуку вразливостей

ISSN 2663 - 4023

Навчальні дані

Ключовим моментом у використанні глибокого навчання є надходження відповідних навчальних даних. Наш підхід до створення навчальних даних для використання в DQN складається з трьох етапів:

- Використання Shodan для збору мережевої інформації та моделювання реалістичного мережевого середовища;
- Використання MulVAL для побудови дерева атак, яке відповідає даному мережевому середовищу;
- Попередня обробка даних з метою їх перетворення на формат, придатний для використання в моделі DQN.

Набір даних хоста: З метою побудови навчальних даних, на першому етапі ми використовуємо інструмент Shodan для збору інформації про реальні мережеві пристрої. Наприклад, якщо нашим завданням є отримання даних про реальні веб-сервери, API Shodan надасть нам відповідну інформацію, таку як фактичну IP-адресу, використовувані порти та протоколи, відомі вразливості тощо. Нижче наведений приклад зібраних даних засобом Shodan (конфіденційна інформація, наприклад, IP-адреси, була вилучена).

```
{
  "info": "(Ubuntu)",
  "ip_str": "192.168.1.2",
  "isp": "HiNet",
  "os": null,
  "port": 80,
  "product": "Nginx",
  "transport": "tcp",
  "version": "1.14.0",
  "vulns": {
    "CVE-2018-16845",
  }
```

Для кожної окремої послуги ми створюємо окремий файл даних про сервіс з усіма відповідними деталями, що стосуються конкретного мережевого хоста, на якому запущено цей сервіс. Під час використання зібраних даних ми забезпечуємо захист конфіденційності, зберігаючи тільки неідентифіковану інформацію, таку як відкриті порти або назви сервісів. У табл. 1 наведено кілька прикладів профілів веб-серверів (конфіденційна інформація, наприклад, IP-адреси, була вилучена).

Таблиця 1

Приклади веб серверів

Продукт	**Порт**	**Протокол**	**Вразливість**	**ОС**
Apache Tomcat	8080	HTTPS	CVE-2021-25329	Debian
Microsoft IIS	443	HTTPS	CVE-2017-7269	Windows 10

Таким чином, ми отримуємо реальний набір даних хостів за допомогою Shodan.

Набір даних вразливостей: Крім набору даних про хоста, ми також створюємо файл з даними про вразливості. Для набору відомих вразливостей, набір даних про вразливості включає CVE та ідентифікатори вразливостей від Microsoft номери вразливостей, а також тип, базову оцінку та можливість експлуатації компоненти оцінки CVSS [9]. З метою забезпечення повної інформації, ми поєднуємо дані з Національної бази даних вразливостей (National Vulnerability Database) (NVD) та бази даних Microsoft

(MS) для створення нового набору даних. У табл. 2 показано кілька прикладів елементів з набору даних про вразливості.

Набір даних DQN: Набір даних DQN — це навчальний набір даних для Deep Q-Learning мережі, що включає в себе набір даних про хост, зібраний з використанням Shodan, та набір даних про вразливості. Для генерації набору даних, необхідного для алгоритму DQN, ми спочатку повинні створити дерево атак для визначеного набору топологій мережі.

Таблиця 2

Приклад набору даних про вразливості

CVE ID	MS ID	Тип	BaseScore	ExpScore
CVE-2015-2426	MS15-034	Помилки буфера	8.5	8.5
CVE-2021-34527	MS21-031	Виконання команд (RCE)	9.8	9.8

Ми створили набір шаблонів топології мережі, що заповнюються реальними даними з датасету пов’язаного із декількома хостами. Ці шаблони мережевих топологій можуть бути використані для визначення конфігурацій мережі. Нижче наведено приклад шаблону для вразливої конфігурації веб-сервера, що включається у датасет.

```
vulExists(webApp, 'CVE-2021-1234', nginx) .
vulProperty('CVE-2021-1234', remoteCodeExec, denialOfService) .
networkServiceInfo(webApp, nginx, tcp, 443, Ubuntu) .
```

З цього прикладу ми можемо зрозуміти, що властивість vulExists відноситься до даних, отриманих за допомогою Shodan, і вказує на те, що метою є веб-сервер з уразливістю CVE-2021-1234. Властивість vulP була взята з набору даних про вразливості та показує, що тип CVE-2021-1234 — віддалене виконання коду, яке може призвести до підвищення привілеїв. Ця детальна інформація про топологію мережі використовується MulVAL для створення дерева атак, яке відповідає цій топології.

Топології мережі та згенеровано дерево атак показані на рис. 4 та рис. 5 відповідно. Після цього дерево атак потрібно перетворити у матрицю переходів. Раніше, Юсефі та ін. [7] запропонували метод перетворення дерева атак у матрицю переходів, але він не підходить для тестування на проникнення, оскільки в процесі тестування можуть бути виявлені інші етапи, які є важливими, такі як доступ до файлів, виконання команд тощо. З цієї причини ми запропонували вдосконалений метод перетворення дерева атак на спрощену матрицю переходів.

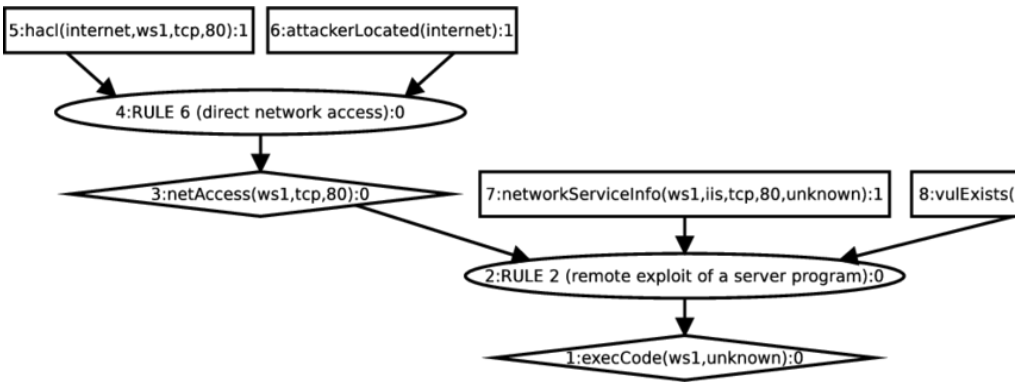

Рис. 4. Дерево атак

Спочатку ми перетворюємо всі вузли дерева атак у матричну форму, яка містить інформацію про оцінку CVSS вразливостей та оцінку для інших дій, таких як доступ до файлів. Далі ми спрощуємо матрицю за допомогою алгоритму пошуку в глибину (Depth-First Search, DFS) замість завантаження її безпосередньо в алгоритм DQN. Ідея полягає в тому, що повна матриця переносу містить всі можливі переміщення, але її можна спростити, якщо вибирати тільки ті шляхи, які можуть бути використані для досягнення мети атаки. Таким чином, ми використовуємо алгоритм DFS для знаходження всіх можливих шляхів, які можуть бути використані для досягнення мети, а потім створюємо спрощену матрицю, яка містить: оцінки для початкового вузла у першому стовпці; сумарні оцінки для проміжних кроків у проміжних стовпцях; оцінку для кінцевої вершини в останньому стовпці. Ці бали будуть використовуватися як винагорода в алгоритмі DQN.

У нашому фреймворку ми використовуємо безперервне навчання моделі DQN, щоб визначити найбільш можливі шляхи атаки. Для цього DQN-моделі подається спрощена матриця, яка була описана раніше, а в якості функції активації ми використовуємо функцію sof tmax. Оптимальний шлях атаки є вихідним значенням DQN-моделі. Під час навчання, агент DQN-моделі представляє зловмисника, а цільове середовище моделюється спрощеною матрицею атак. Зловмисник переміщується від вузла до вузла в матриці атаки досягнення цільового сервера. Винагорода, що відповідає використанню кожної вразливості в моделі DQN, обчислюється за допомогою балу вразливості, який був визначений на основі компонентів Загальноприйнятої системи оцінювання вразливостей (CVSS).

$$\text{Score}_{\text{vul}} = {}_{\text{baseScore}} \times \frac{exploitabilityScore}{10}, \quad (1)$$

У системі оцінювання вразливостей CVSS базова оцінка відображає потенційну шкідливість самої вразливості, тоді як оцінка експлуатованості відображає можливість використання даної вразливості. Тому ми використовуємо оцінку можливості використання з максимальним значенням 10 для визначення значущості базового балу в залежності від того, наскільки легко можна використати вразливість.

Інструменти проникнення

Для того, щоб наш автоматизований фреймворк дозволяв проводити атаки на реальні системи, необхідно, щоб він міг взаємодіяти з мережевим середовищем, зокрема, виконувати команди, експлуатувати вразливості тощо. На даний момент ми вирішили використовувати існуючі утиліти, такі як Metasploit [6], за аналогією з підходом, що використовується в CyPROM [2]. Наш підхід полягає в створенні обгортки для інструментів тестування на проникнення, щоб результат моделі DQN, навченої, як описано вище, можна було використовувати для надсилання команд цим інструментам, які будуть виконувати дії на реальних цільових системах. Результати цих дій отримуються у вигляді зворотного зв’язку, який використовується для прийняття рішення щодо подальших дій на заданому шляху атаки.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Ми здійснили серію експериментів для перевірки ефективності нашого підходу, моделюючи топологію мережі і використовуючи алгоритм DQN для знаходження найкращого шляху атаки для цієї топології.

ISSN 2663 - 4023

А. Сценарій експерименту

Рис. 5 демонструє модель топології мережі, яку ми використовували для проведення експериментів. Модель відображає компанію з маленькою мережею, що включає веб-сервер, файловий сервер та робочу станцію. Файловий сервер та робоча станція розташовані в одній підмережі, з'єднаній через маршрутизатор, який, далі, підключений до брандмауера. Веб-сервер знаходиться в іншій підмережі та підключений через брандмауер до Інтернету.

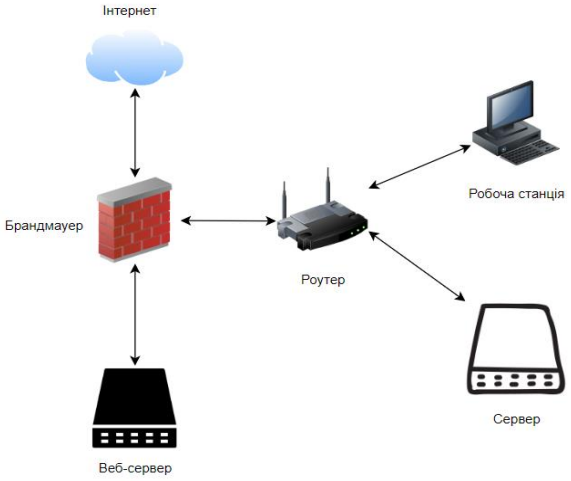

Рис. 5. Топологія мережі для експерименту

З відношення до атак, ми припускаємо наступне:

- Зловмисники розпочинають атаку з Інтернету та можуть отримати доступ до веб-сервера за допомогою протоколів HTTP та HTTPS.
- Файловий сервер та веб-сервер можуть бути підключені один до одного за допомогою файлових протоколів, таких як NFS, FTP тощо.
- Файловий сервер та робоча станція можуть бути підключені один до одного за допомогою файлових протоколів, таких як NFS, Samba тощо.
- Файловий сервер та робоча станція можуть отримати доступ до Інтернету за допомогою протоколів HTTP та HTTPS.

Для уточнення параметрів протоколу ми використали дані, отримані з допомогою Shodan, щоб ініціалізувати інформацію про вразливості, відкриті порти, продукти та протоколи, що використовуються веб-сервером та файловим сервером. Щодо робочої станції, ми вважали, що вона не запускає жодних сервісів, та використовує різні протоколи передачі даних. У табл. 3 представлений приклад мережевої інформаційної конфігурації для цих хостів.

Таблиця 3

Інформація про конфігурацію хоста

Хост	**Вразливість**	**Порт**	**Продукт**	**Протокол**
Веб-сервер	CVE-2010-2730	80	Microsoft IIS	HTTP/HTTPS
Сервер	CVE-2015-3306	21	-	FTP
Робоча станція	-	-	-	HTTPS/FTP

Набір даних щодо вразливостей використовується для надання детальної інформації про кожну вразливість, зокрема її тип та наслідки (наприклад, рівень дозволу, досяжного за допомогою цієї вразливості). Для кожного типу вразливостей, алгоритми дерева атак генерують різні шляхи атаки. У табл. 4 представлені деякі приклади типів вразливостей, які ми зустрічаємо у нашому сценарії.

Таблиця 4

Детальна інформація про вразливості

Вразливість	**Тип**	**Вплив**
CVE-2023-32233	Ескалація привілегій	root
CVE-2023-22515	Проблеми з контролем доступу	user

B. Результати навчання DQN

Спрощена матриця, яку ми розглядали вище, використовується як вхідні дані для DQN моделі, яка потім навчається визначати сумарну винагороду для всіх можливих шляхів. Для топології мережі, яку ми використовували, ми зібрали дані Shodan з різними вразливостями для розглянутих протоколів у навчальному наборі, що дозволило створити загалом декілька сотень різних дерев атак для навчання моделі та сотні різних дерев атак для перевірки. Для валідації ми визначили, що оптимальний шлях у спрощеній матриці — це шлях з найменшою кількістю кроків і найбільшою винагородою. Точність моделі та середня кількість кроків для оптимального шляху атаки представлені в табл. 5. З цих результатів видно, що для цієї топології мережі, модель має високу продуктивність у пошуку найоптимальнішого шляху.

Таблиця 5

Точність і середня кількість кроків

Фреймворк	**Точність**	**Середня кількість кроків**
DQN	0.835	1.877

На рис. 6 показано середні зміни винагороди за моделлю DQN для одного з шляхів атаки у сценарії експерименту протягом 100 навчальних ітерацій. Можна помітити, що винагорода спочатку невелика, а потім поступово зростає. Цей шлях не тільки повністю використовує вразливості як веб-сервера, так і файлового сервера, але й застосовує прості методи атаки які легко перевірити за допомогою інструментів тестування на проникнення.

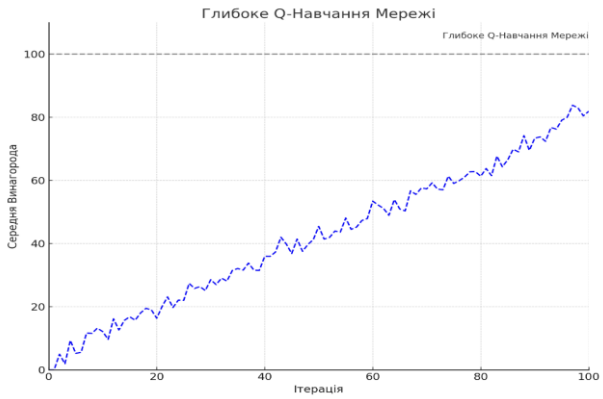

Рис. 6. Зміна винагороди за один шлях атаки в експериментальному сценарії

Наш фреймворк для автоматизованого тестування на проникнення реальних систем було розроблено, але до цього моменту ми проводили навчання на загальних моделях мережевої топології. Наступним кроком є розгляд можливості застосування алгоритму DQN для навчання моделі на основі результатів сканування конкретних реальних топологій мереж. Це дозволить оцінити конкретні характеристики цієї топології та запропонувати оптимальний шлях атаки для неї. Кожен вузол на найбільш ймовірному шляху атаки, визначеному алгоритмом DQN, містить всю інформацію, необхідну для ефективного проведення цієї атаки. Інформація про шлях та вузли може бути використана для керування виконанням інструменту тестування на проникнення, наприклад, Metasploit.

Наприклад, якщо фреймворк знає, що вузол має певну вразливість, він може відправити команду до Metasploit та використати модуль msf для експлуатації цієї вразливості на веб-сервері. Наш фреймворк був розроблений для використання у проведенні навчання з кібербезпеки, його можна використовувати для тренувальних сценаріїв, де слухачам пропонуються шляхи атак, з якими вони можуть експериментувати самостійно в кіберпросторі, дотримуючись керованої методології навчання. Повторення тренувальних заходів з використанням реалістичних методів атак значно покращить навички захисту учасників.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Ми розробили фреймворк для автоматизованого тестування на проникнення на основі методів глибокого навчання, зокрема Deep Q-Learning Network (DQN), і поєднали систему пошуку Shodan з різними даними про вразливості для збору реальних даних про хости та вразливості для створення реалістичних навчальних та тестових сценаріїв. За допомогою дерева атак ми генеруємо інформацію про атаки для кожного сценарію, яка потім використовується для навчання моделі DQN. Винагорода використовується для оцінки кінцевого шляху атаки, в основному, використовуючи оцінки CVSS. Ми випробували нашу систему на топології мережі, заповненій реальними даними про хости та вразливості. І навіть з такою малою кількістю навчальних сценаріїв, DQN досяг досить високої точності, при визначенні оптимального шляху атаки і в інших випадках давав раціональні рішення. Наш фреймворк може бути використаний для створення стратегій атак та підтримки в проведенні тренінгів з кібербезпеки.

Наша розробка фреймворку для автоматизованого тестування на проникнення з використанням глибокого навчання з підкріпленням відкриває нові перспективи для подальших досліджень у цій області.

1. Покращення алгоритмів глибокого навчання з підкріпленням: Існує потенціал для розвитку більш складних алгоритмів DQN або використання інших методів глибокого навчання з підкріпленням, таких як Policy Gradient або Actor-Critic алгоритми, що можуть підвищити ефективність виявлення та експлуатації вразливостей.
2. Інтеграція з іншими інструментами безпеки: Інтеграція нашого фреймворку з іншими інструментами та платформами безпеки, такими як SIEM системи (Security Information and Event Management), може забезпечити більш комплексний підхід до кібербезпеки.
3. Адаптація до змінюваних загроз: Враховуючи швидкий розвиток кіберзагроз, важливо адаптувати фреймворк до нових видів атак та вразливостей. Це може

включати навчання моделі на нових даних та регулярне оновлення бази даних вразливостей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Academic: Attack Trees - Schneier on Security.* (n.d.). Schneier on Security. https://www.schneier.com/academic/archives/1999/12/attack_trees.html
2. *Cyprom - Pentesting Project Management :: Gonkar IT security.* (n.d.). Gonkar IT Security :: Cybersecurity Services. https://gonkar.com/cyprom
3. *The Dark Side Of The Internet: A Search Engine That Finds Unsecured Routers, Servers & A Whole Lot More.* (n.d.). Search Engine Land. https://searchengineland.com/the-dark-side-of-the-internet-a-search-engine-that-finds-unsecured-routers-servers-a-whole-lot-more-154943
4. *HIPAA home.* (n.d.). HHS.gov. https://www.hhs.gov/hipaa/index.html
5. Hoffmann, J. (2015). Simulated penetration testing: From "dijkstra" to "turing test++". *Proceedings of the International Conference on Automated Planning and Scheduling, 25,* 364–372. https://doi.org/10.1609/icaps.v25i1.13684
6. *Metasploit|Penetration Testing Software, Pen Testing Security|Metasploit.* (n.d.). Metasploit. https://www.metasploit.com/
7. Yousefi, M., et al. (2018). A reinforcement learning approach for attack graph analysis. *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).* https://doi.org/10.1109/TrustCom/BigDataSE.2018.00041
8. Nguyen, T., & Reddi, V. (2021). Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems,* 1–17. https://doi.org/10.1109/tnnls.2021.3121870
9. *NVD - CVSS v3 Calculator.* (n.d.). NVD - Home. https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator
10. *Official PCI security standards council site.* (n.d.). PCI Security Standards Council. https://www.pcisecuritystandards.org/
11. Oriyano, S.-P. (2017). Penetration testing essentials. *Wiley & Sons, Limited, John.*
12. Sembiring, J., et al. (2015). Network security risk analysis using improved mulval bayesian attack graphs. *Int. J. Electrical Eng. Inf. 7(4),* 735–753. https://doi.org/10.15676/ijeei.2015.7.4.15
13. *What is Shodan? The search engine for everything on the internet.* (n.d.). CSO Online. https://www.csoonline.com/article/565528/what-is-shodan-the-search-engine-for-everything-on-the-internet.html

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (23), 2024
ISSN 2663 - 4023

Anastasiia Tolkachova
Cybersecurity student
Lviv Polytechnic National University, Lviv, Ukraine
ORCID ID 0000-0002-8196-7963
anastasiia.tolkachova.mkbst.2022@lpnu.ua

Maksym-Mykola Posuvailo
Cybersecurity student
Lviv Polytechnic National University, Lviv, Ukraine
ORCID ID 0009-0000-8553-2142
posuvaylom@gmail.com

PENETRATION TESTING USING DEEP REINFORCEMENT LEARNING

Abstract. Traditionally, penetration testing is performed by experts who manually simulate attacks on computer networks to assess their security and identify vulnerabilities. However, recent research highlights the significant potential for automating this process through deep reinforcement learning. The development of automated testing systems promises to significantly increase the accuracy, speed and efficiency of vulnerability detection and remediation. In the pre-testing phase, artificial intelligence can be used to automatically create a realistic network topology, including the development of a tree of possible attacks. The use of deep learning methods, such as Deep Q-Learning, allows the system to determine the best attack paths, making the penetration process more strategic and informed. Automated penetration testing systems can serve as effective training tools for cybersecurity professionals. They allow attacks to be simulated in a controlled training environment, providing users with the opportunity to analyse different intrusion strategies and techniques, and serve as a training tool for detecting and responding to real-world attacks. This approach promotes a deep understanding of potential threats and develops the skills to effectively defend against them. In addition, the use of machine learning can help solve the problem of large numbers of false positives, which is a common problem in traditional security systems. Deep reinforcement learning offers the opportunity to create more adaptive scanning systems that can learn and adapt to changing threat patterns. Such systems are not only more efficient, but also able to operate with fewer errors, reducing the burden of human error. As a result, they can identify vulnerabilities that humans may not, providing a deeper and more comprehensive security analysis. This approach has the potential to revolutionise the cybersecurity industry, offering new strategies for protecting information systems and creating more robust network structures.

Keywords penetration testing; artificial intelligence; machine learning; reinforcement learning; network security audit; offensive cybersecurity; vulnerability assessment.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. *Academic: Attack Trees - Schneier on Security.* (n.d.). Schneier on Security. https://www.schneier.com/academic/archives/1999/12/attack_trees.html
2. *Cyprom - Pentesting Project Management :: Gonkar IT security.* (n.d.). Gonkar IT Security :: Cybersecurity Services. https://gonkar.com/cyprom
3. *The Dark Side Of The Internet: A Search Engine That Finds Unsecured Routers, Servers & A Whole Lot More.* (n.d.). Search Engine Land. https://searchengineland.com/the-dark-side-of-the-internet-a-search-engine-that-finds-unsecured-routers-servers-a-whole-lot-more-154943
4. *HIPAA home.* (n.d.). HHS.gov. https://www.hhs.gov/hipaa/index.html
5. Hoffmann, J. (2015). Simulated penetration testing: From “dijkstra” to “turing test++”. *Proceedings of the International Conference on Automated Planning and Scheduling, 25,* 364–372. https://doi.org/10.1609/icaps.v25i1.13684
6. *Metasploit|Penetration Testing Software, Pen Testing Security|Metasploit.* (n.d.). Metasploit. https://www.metasploit.com/

7. Yousefi, M., et al. (2018). A reinforcement learning approach for attack graph analysis. *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).* https://doi.org/10.1109/TrustCom/BigDataSE.2018.00041
8. Nguyen, T., & Reddi, V. (2021). Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems*, 1–17. https://doi.org/10.1109/tnnls.2021.3121870
9. *NVD - CVSS v3 Calculator*. (n.d.). NVD - Home. https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator
10. *Official PCI security standards council site*. (n.d.). PCI Security Standards Council. https://www.pcisecuritystandards.org/
11. Oriyano, S.-P. (2017). Penetration testing essentials. *Wiley & Sons, Limited, John*.
12. Sembiring, J., et al. (2015). Network security risk analysis using improved mulval bayesian attack graphs. *Int. J. Electrical Eng. Inf. 7(4)*, 735–753. https://doi.org/10.15676/ijeei.2015.7.4.15
13. *What is Shodan? The search engine for everything on the internet.* (n.d.). CSO Online. https://www.csoonline.com/article/565528/what-is-shodan-the-search-engine-for-everything-on-the-internet.html