DOI: 10.58254/viti.4.2023.09.100

УДК 004.49+004.8

д-р техн. наук, ст. наук. співроб. Чевардін В. Є. ORCID: 0000-0002-1070-4568 (ВІТІ ім. Героїв Крут)
канд. техн. наук Юрченко О. В. ORCID: 0000-0002-4851-0400 (ВІТІ ім. Героїв Крут)
канд. техн. наук Залужний О. В. ORCID: 0000-0002-8722-4087 (ВІТІ ім. Героїв Крут)
канд. техн. наук Пелешок Є. В. ORCID: 0000-0003-0033-1160 (НДІ ВР)

АНАЛІЗ КОНКУРЕНТНИХ АТАК НА МОДЕЛІ МАШИННОГО НАВЧАННЯ СИСТЕМ КІБЕРЗАХИСТУ

Сучасні тенденції розвитку систем кіберзахисту пов'язані з широким застосуванням технологій машинного навчання для виявлення і запобігання кіберзагрозам. Водночас зловмисники шукають способи ухилення від детектування такими системами, використовуючи при цьому як традиційні методи атак, так і нові, орієнтовані виключно на протидію штучному інтелекту, – конкурентні атаки. Тому пошук шляхів протидії конкурентним атакам є актуальним науково-технічним завданням. Для їх вивчення використовують конкурентне машинне навчання (Adversarial Machine Learning – AML), яке полягає у моделюванні таких атак.

Метою досліджень є визначення шляхів підвищення стійкості систем кіберзахисту, що функціонують із використанням технологій машинного навчання, до впливу атак на основі AML-моделей.

У статті наведено приклади застосування методів машинного навчання в системах кіберзахисту. Проведено опис моделей конкурентних атак, а саме: моделі ухилення, отруєння, функціонального вилучення, інверсії та моделі атаки на приналежність. Розглянуто можливі сценарії їхнього здійснення. Проаналізовано приклади конкурентних атак на моделі машинного навчання для розпізнавання зображень та текстових повідомлень, виявлення алгоритмів генерації доменних імен, шкідливого програмного забезпечення в HTTP-трафіку, шкідливого вмісту в електронних листах, обходу антивірусних програмних засобів.

Дослідження показали, що навіть не маючи доступу до алгоритмів роботи моделей машинного навчання, можливо реалізувати обхід системи кіберзахисту. Тому для забезпечення безпеки мереж і послуг засобами кіберзахисту зі штучним інтелектом необхідно враховувати необхідність протидії конкурентним атакам. Із цією метою запропоновано: здійснювати збирання та агрегацію навчальних даних для кожної моделі машинного навчання окремо, а не отримання їх із загально-доступних джерел; проводити оптимізацію вмісту журналів подій, з урахуванням можливості використання інформації, що знаходиться в них для створення конкурентних атак; забезпечувати захист навчальних даних та алгоритмів функціонування моделей; у випадку розгортання систем кіберзахисту на об'єктах критичної інфраструктури використовувати спеціально розроблені моделі машинного навчання, яких немає в загальному доступі, що ускладнить можливість створення атаки функціонального вилучення.

Ключові слова: *штучний інтелект, машинне навчання, глибоке навчання, конкурентне машинне навчання, конкурентні атаки, шкідливе програмне забезпечення, кібератаки, кіберзахист, кібербезпека.*

V. Chevardin, O. Yurchenko, O. Zaluzhnyi, Ye. Peleshok Analysis of adversarial attacks on the machine learning models of cyberprotection systems.

Modern trends in the development of cyber protection systems are associated with the widespread use of machine learning technologies to detect and prevent cyber threats. At the same time, attackers are looking for ways to evade detection by such systems, using both traditional attack methods and new ones aimed exclusively at countering artificial intelligence - adversarial attacks. Therefore, finding ways to counteract adversarial attacks is an urgent scientific and technical task. Adversarial Machine Learning (AML) is used to study them, which consists in simulating such attacks.

The purpose of research is to determine ways to increase the resilience of cyber defense systems operating with the use of machine learning technologies to the impact of attacks based on AML models.

The article provides examples of the application of machine learning methods in cyber protection systems. The models of adversarial attacks are described, namely: models of evasion, poisoning, functional extraction, inversion, and models of membership inference attack. Possible scenarios of their implementation are considered. Examples of adversarial attacks on machine learning models for recognizing images and text messages, detecting domain name generation algorithms, HTTP traffic malware, malicious content in e-mails, bypassing antivirus software are analyzed.

Studies have shown that even without access to the algorithms of machine learning models, it is possible to bypass the cyber protection system. Therefore, to ensure the security of networks and services by means of cyber protection with artificial intelligence, it is necessary to take into account the need to counter adversarial attacks. For this purpose, it is proposed to: collect and aggregate training data for each machine learning model individually, instead of obtaining them from publicly available sources; optimize the content of event logs, taking into account the possibility of using the information contained in them to create adversarial attacks; to ensure the protection of training data and algorithms of

the functioning of models; in the case of deploying cyber protection systems on critical infrastructure objects, use specially developed machine learning models that are not publicly available, which will complicate the possibility of creating a functional extraction attack.

Keywords: *artificial intelligence, machine learning, deep learning, adversarial machine learning, adversarial attacks, malware, cyberattacks, cyber defense, cyber security.*

Постановка завдання в загальному вигляді. В наш час штучний інтелект (*Artificial Intelligence – AI*) активно застосовується в системах кіберзахисту. Машинне навчання (*Machine Learning – ML*) є компонентом *AI*. Одним із напрямків досліджень у сфері *ML* є глибоке навчання (*Deep Learning – DL*). Системи кіберзахисту на основі *ML* здійснюють детектування кібератак, шкідливого програмного забезпечення (далі – ШПЗ), виявлення аномалії в мережі та ін. [1; 2], однак *ML*-моделі також є об'єктом кібератак. Атаки, що призводять до прийняття хибних рішень *ML*-моделлю, називають конкурентними (*adversarial attacks*) [3]. Вивченням можливостей зловмисників і їхніх цілей, а також розробкою методів атак, що експлуатують вразливості *ML*-моделей на етапах розробки, навчання і використання, займається конкурентне машинне навчання (*Adversarial Machine Learning – AML*) [4; 5].

Враховуючи, що майбутній розвиток систем кіберзахисту пов'язаний із широким застосуванням моделей *ML*, гострим та актуальним науково-технічним завданням є дослідження існуючих *AML*-моделей та пошук шляхів протидії конкурентним атакам.

Аналіз останніх публікацій

Значна кількість досліджень із кібербезпеки, присвячених застосуванню методів машинного навчання, свідчить про їхню вагому роль у сфері кіберзахисту інформаційно-комунікаційних систем [1; 2; 6–10]. Зокрема, дерево рішень використовується для виявлення вторгнень [1; 6]. Для виявлення аномальної поведінки IoT (Internet of Things) пристроїв застосовується метод опорно-векторних машин та *k*-найближчих сусідів [8]. Останній також знаходить місце при виявленні фішингових атак [7]. Наївний баєсів класифікатор застосовується для виявлення аномального вмісту мережевих пакетів [9]. Логістична регресія дозволяє виявляти шкідливий ботнет-трафік [10]. Однак ці інструменти, за своєю суттю, не є надійними та безпечними. Зловмисники, які хочуть уникнути виявлення незахищеними моделями машинного навчання, можуть зробити це з відносною легкістю. В існуючих наукових публікаціях наведено перелік відомих атак на моделі машинного навчання та їхню класифікацію [11], розглянуто основні аспекти безпеки технологій машинного навчання та напрямки здійснення атак [12], але не проведено аналіз прикладів успішних атак на системи кіберзахисту, що функціонують на основі штучного інтелекту.

Метою статті є визначення шляхів підвищення стійкості систем кіберзахисту, що функціонують із використанням технологій машинного навчання, до впливу атак на основі *AML*-моделей.

Виклад основного матеріалу

Відомими методами детектування подій у кіберпросторі є сигнатурний та поведінковий аналізи. Алгоритми на основі сигнатурного аналізу не забезпечують захист від використання поліморфних кодів та диверсифікації ШПЗ. Для розв'язання цих задач, як правило, застосовують поведінковий аналіз на основі *ML*-моделей. Для таких систем зловмисники розробляють свої методи, прийоми та способи обходу. Відомі приклади атак описані в матриці *MITRE ATLAS™* (*Adversarial Threat Landscape for Artificial-Intelligence Systems*) [13] – базі знань про прийоми і методи дій проти моделей машинного навчання. ATLAS було розроблено з метою підвищення обізнаності фахівців щодо наявних загроз. До неї увійшли дані, отримані з досвіду *IBM, NVIDIA, Bosch, Microsoft* та інших відомих компаній, які відіграють важливу роль у сфері інформаційних технологій. Наведена матриця надає аналітикам безпеки систематизовану картину загроз для *ML*-моделей.

У процесі створення будь-якої моделі машинного навчання здійснюється збирання, аналіз та оброблення даних, навчання, перевірка роботи моделі та її впровадження. Після

впровадження користувачі мають змогу надсилати запити та отримувати відгуки (результати роботи) *ML*-моделі. Відповідно, існують два напрямки здійснення шкідливого впливу на *ML*-моделі: вплив на етапі навчання моделі (перед впровадженням) і вплив на етапі її використання (на етапі отримання відгуку моделі). На цих етапах можуть бути реалізовані різні моделі атак, а саме: модель ухилення (*evasion model*), модель отруєння (*poisoning model),* модель функціонального вилучення (*functional extraction model),* модель інверсії (*inversion model*) та модель атаки на приналежність (*model of membership inference attack*) [4]. У таблиці 1 наведено опис цих моделей та вказано етапи, на яких вони застосовуються.

Таблиця 1

***AML*-моделі атак**

Модель атаки	Опис атаки	Етап атаки
Модель ухилення (*Evasion model*)	Зловмисник змінює запит до моделі, щоб отримати бажаний результат (обійти захист). Для цього йому потрібно вивчити функціонування моделі, навіть не знаючи її алгоритмів. Такі атаки виконуються шляхом надсилання різних за змістом запитів до моделі та спостереження за результатом (відгуком моделі)	Використання
Модель отруєння (*Poisoning model*)	Зловмисник отримує доступ та змінює навчальні дані *ML*-моделі або саму модель, щоб отримати бажаний результат її роботи. Модель може бути «перепрограмована» для виконання нового непередбаченого розробниками завдання. Доступ до навчальних даних також може призвести до компрометації особистих даних користувачів	Навчання
Функціональне вилучення (*Functional Extraction model*)	Зловмисник створює (відтворює) функціонально еквівалентну модель (офлайн-копію моделі) шляхом ітераційних запитів до моделі *ML* та оцінки відгуків. Це дозволяє зловмиснику перевірити створену офлайн-копію моделі перед подальшою атакою на онлайн-модель (робоча модель)	Використання
Модель інверсії (*Inversion model*)	На основі аналізу відгуків моделі зловмисник здійснює прогнозування вхідних даних цієї моделі. Аналізуючи ці дані зловмисник може дізнатись інформацію про суб'єкт даних	Використання
Модель атаки на приналежність *(Model of membership inference attack)*	Зловмисник визначає, чи є вказаний запис даних частиною набору даних для навчання моделі. Виявлення таких даних може призвести до проблем із конфіденційністю у випадках, якщо модель навчали, використовуючи конфіденційну інформацію	Використання

Розглянуті моделі атак можуть бути реалізовані за наступними сценаріями.

Сценарій 1. Атака на основі відгуку ML-моделі (Inference Attack) (рис. 1). Це найпоширеніший сценарій, при якому зловмисник може лише надсилати запити до моделі і спостерігати за її відповіддю (модель розгортається як кінцева точка *API – Application Programming Interface*). Зловмисник контролює вхідні дані в модель, але він не знає, як ці дані обробляються [14].

Атаки за цим сценарієм мають за мету створення такого набору даних, при обробці якого модель видаватиме хибні результати. Один із результатів його виконання описано в [15]. На працездатність була перевірена *ML* модель розпізнавання образів, що використовує нейронні мережі. В наведеному прикладі було застосовано метод швидкого градієнта (*Fast Gradient Sign Method – FGSM*) для генерації конкурентних (*adversarial*) вхідних даних (зображень) та здійснено їх тестування на згортковій нейронній мережі "*GoogLeNet*".

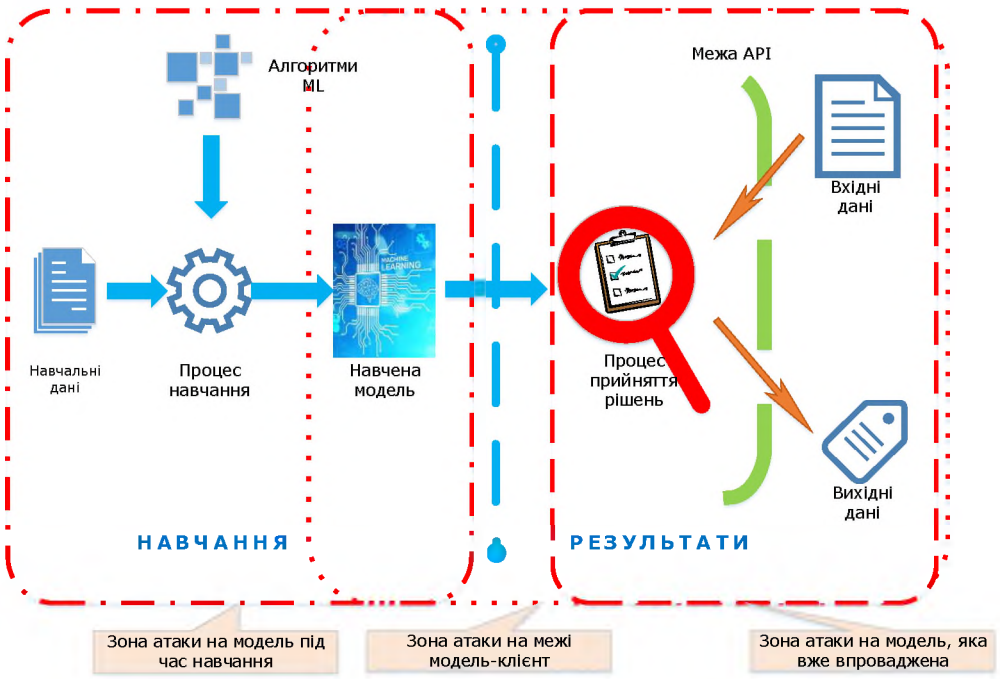

Рис. 1. Зони атак на моделі машинного навчання

Суть методу полягає в тому, що зловмисник модифікує вихідне зображення в напрямку градієнта функції втрат відносно вхідного зображення. Значення змін має бути настільки малим, щоб не бути детектованим. Якщо розмір конкурентної пертурбації $\|\eta\|_\infty < \varepsilon$ (де $\|\eta\|_\infty$ – максимальна або нескінченна норма), то конкурентний зразок можна обчислити за формулою (1) [15]:

$$\hat{x} = x + \varepsilon \cdot \text{sign}\left(\nabla_x J(\theta, x, y)\right), \tag{1}$$

де x – вхідне зображення;

$J(\theta, x, y)$ – функція втрат;

θ – параметри моделі;

Y – зображення, яке має розпізнати згорткова мережа;

∇_x – градієнт функції втрат.

У наведеному прикладі значення максимально-допустимих змін, що вносяться в зображення, $\varepsilon = 0{,}007$.

На рисунку 2 продемонстровано застосування методу *FGSM*.

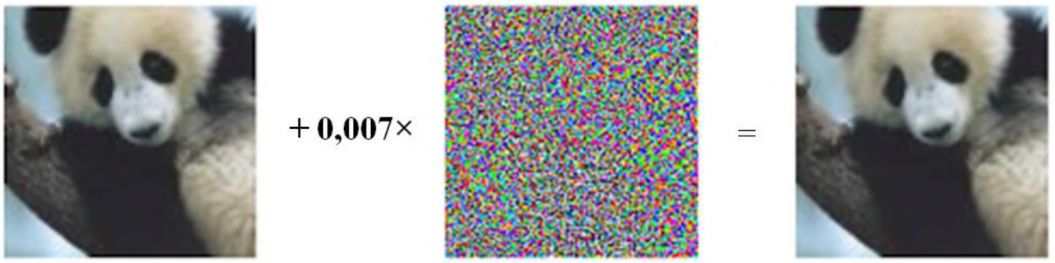

Рис. 2. Приклад виконання атаки на впроваджену модель розпізнавання зображень

Спочатку модель розпізнала панду на фотографії з ймовірністю 57,7 %. Потім до зображення додали незначний шум, створений за алгоритмом *FGSM*. Після накладання зображень модель з ймовірністю 99,3 % розпізнала панду як гібона, хоча людина безпомилково визначила б зображений об'єкт як панду.

Сценарій 2. Атака на ML-модель під час її навчання (Training Time Attack) (див. рис. 1*)*. У цьому випадку зловмисник контролює навчальні дані, отримавши попередньо доступ до них [14]. Один із результатів виконання такого сценарію описано в [16]. Корпорація *Microsoft* створила *Tay*, чат-бот у Twitter, для молоді США. Він був розроблений, щоб охопити широкі кола молоді з її специфічним сленгом. *Tay* використовував взаємодію з користувачами *Twitter* як навчальні дані для покращення своєї лексики та розширення словарного запасу. Розробники сподівались, що властивість боту накопичувати необхідну лексичну інформацію призведе до покращення рівня спілкування з користувачами. Група користувачів *Twitter* об'єдналася з наміром зіпсувати бот *Tay*, використовуючи такий зворотний зв'язок. У своїх скоординованих зусиллях вони використали функцію «повторюй за мною», яка була вбудована в *Tay*. Вже через 16 годин чат-бот створив 95 000 повідомлень і вони були переважно лайливими та образливими. Внаслідок цієї скоординованої атаки навчальні дані *Tay* були отруєні і *Microsoft* зупинила дію свого чат-бота. Те, що починалося цікавим експериментом, менш ніж за добу зазнало краху.

Сценарій 3. Атака на межі ML-модель – клієнт (Attack on Edge/Client) (див. рис. 1). У цьому випадку модель встановлена у клієнта (наприклад, у телефоні) або використовується на межі модель – клієнт (наприклад, інтернет-речей). Зловмисник може оцінити алгоритми функціонування моделі шляхом застосування методів реверс-інжинірингу до служби, встановленої у клієнта [14]. Приклад атаки за цим сценарієм буде наведено нижче.

Розглянуті сценарії ілюструють атаки за допомогою «чорної скриньки». Також ці сценарії атак використовуються при налаштуванні «білої скриньки», коли зловмисник отримує доступ до архітектури *ML*-моделі, вихідного коду або навчальних даних. Подібні підходи застосовуються і під час здійснення атак на *ML*-моделі систем кіберзахисту.

Розглянемо декілька прикладів успішних атак на системи кіберзахисту, які використовують штучний інтелект.

Ухилення від детектування ШПЗ в HTTP-трафіку Deep Learning моделлю.

Дослідницька група *Palo Alto Networks Security AI* випробувала *DL*-модель для виявлення трафіку контролю та управління (*C*2-трафік) шкідливим програмним забезпеченням у *HTTP*-трафіку (робоча модель). Вказана модель була запропонована в роботі: «URLNet: Learning a URL representation with deep learning for malicious URL detection» [17]. Проаналізувавши статтю, дослідники створили функціонально еквівалентну модель (ФЕМ) та навчили її на датасеті *C*2-трафіку *HTTP*-протоколу, що містив близько 33 мільйонів нешкідливих і 27 мільйонів шкідливих заголовків *HTTP*-пакетів. Оцінка моделі показала близько 99 % *«true positive»* результатів, при цьому *«false positive»* результатів було менше 1 %.

Тестування ФЕМ здійснювалось на заголовках *HTTP*-пакетів відомих зразків *C*2-трафіку ШПЗ. Достовірність виявлення ШПЗ перевищила 99 %. Наступним кроком було створення зразків ухилення від детектування шляхом видалення полів із заголовка пакета, які зазвичай не використовуються для передачі *C*2-трафіку (наприклад, керування кешем, встановлення з'єднання тощо). Отримані зразки тестувались на розробленій моделі та коригувались доти, доки не було забезпечено ухилення від виявлення.

За допомогою створених зразків було виконано онлайн-ухилення від робочої моделі виявлення ШПЗ. Створені пакети були визначені як доброякісні з достовірністю >80 % [18]. Загальна схема ухилення зображена на рисунку 3.

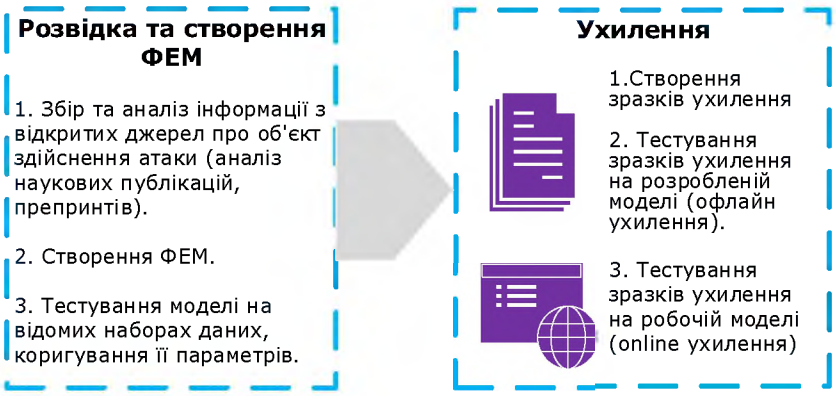

Рис. 3. Схема ухилення від детектування ШПЗ в *HTTP*-трафіку

Ухилення від виявлення доменних імен, що створені з використанням Domain Generation Algorithm (DGA).

Дослідницька група *Palo Alto Networks Security AI* змогла обійти детектор, який здійснює виявлення доменних імен, що створені з використанням *Domain Generation Algorithm* (*DGA*) [19]. Спочатку дослідники провели пошук наукових праць та технічних рішень на основі штучного інтелекту, що пов'язані з виявленням *DGA*. Далі вони протестували загальнодоступну модель виявлення *DGA, що базується* на згортковій нейронній мережі (*Convolutional Neural Network – CNN*), на наборі даних доменних імен (створених *DGA*), який містить 50 мільйонів доменних імен із 64 сімейств ботнетів. Точність виявлення ботнетів склала більш ніж 70 % на 16 сімействах ботнетів (25 %). На наступному кроці, скориставшись результатами наукових публікацій [20], дослідники розробили техніку «мутації» доменних імен. Внаслідок експерименту було виявлено, що після того, як у доменне ім'я, згенероване *DGA*, одноразово вставили лише один рядок, рівень виявлення всіх 16 сімейств ботнетів *DGA* впав до менш ніж 25 %.

Запропонована дослідниками техніка «мутації» дозволяє уникати виявлення *DGA ML*-моделями, не обмежуючись обходом засобів захисту на основі *CNN* [19]. Загальна схема ухилення зображена на рисунку 4.

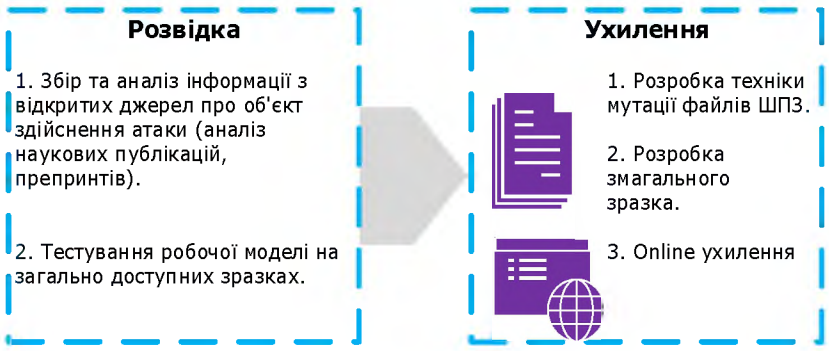

Рис. 4. Схема ухилення від виявлення DGA ботмереж у доменних іменах

Масоване отруєння (Poisoning).

Дослідниками компанії McAfee було помічено незвичне збільшення кількості звітів про відоме сімейство програм-вимагачів [21]. Під час розслідування справи було виявлено, що багато зразків цього сімейства були подані через популярну платформу обміну вірусами протягом короткого проміжку часу (VirusTotal). Подальше дослідження показало, що на основі подібності рядків усі зразки були еквівалентними, а на основі подібності коду вони

були схожими на 74–98 %. Цікаво, що час компіляції був однаковий для всіх зразків. Після додаткових досліджень було встановлено, що зловмисники використовували «metame» – інструмент маніпулювання метаморфічним кодом, щоб маніпулювати вихідними (не шкідливими) файлами для створення «мутантних» варіантів [22]. Створені варіанти не завжди були виконуваними файлами, але *ML*-модель класифікувала їх як одне й те саме сімейство програм-вимагачів.

Послідовність дій зловмисника була такою: крок 1 – використання зразка зловмисного програмного забезпечення з поширеного сімейства програм-вимагачів як основи для створення «мутантних» варіантів; крок 2 – завантаження на платформу VirusTotal зразків «мутантів». Як наслідок, системи захисту почали класифікувати файли як сімейство програм-вимагачів, хоча більшість із цих файлів навіть не запускались. Зразки «мутантів» отруїли набір даних, які *ML*-модель використовує для ідентифікації та класифікації цього сімейства програм-вимагачів. Схема масованого отруєння зображена на рисунку 5.

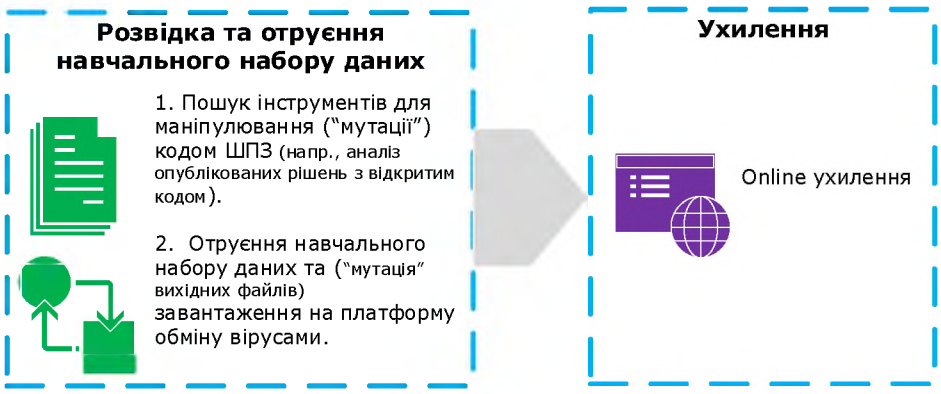

Рис. 5. Схема масованого отруєння моделі детектування програм-вимагачів

Обхід виявлення ШПЗ системою «Cylance AI».

Дослідники компанії *SkylightCyber* [13] змогли знайти універсальний спосіб обходу антивірусного програмного засобу (АВПЗ) «Cylance», який функціонує на основі штучного інтелекту. Компанією було проведено дослідження моделі штучного інтелекту, що використовується в АВПЗ «Cylance». Спочатку було здійснено емпіричне тестування різних нешкідливих і шкідливих файлів та визначено, що оцінка може коливатися від –1000 для найбільш шкідливих файлів до +1000 для найбільш безпечних файлів. Надалі, з метою визначення механізму підрахунку балів, для подальшого обходу було здійснено реверс інжиніринг коду програми. Проаналізовано процес вилучення ознак з виконуваних файлів PE (Portable Executable) формату та особливості формування вектору ознак. На основі проведених досліджень вдалось сформувати список рядків, які необхідно додати до шкідливого файлу для того, щоб значно зменшити ймовірність його детектування. З використанням розробленого способу 88,4 % модифікованих шкідливих файлів було детектовано як нешкідливі. Схему дій зловмисників зображено на рисунку 6.

Обхід виявлення шкідливого вмісту в електронному листуванні.

Випадок CVE-2019-20634 описує [19], як дослідники із *Silent Break Security* змогли ухилитися від системи захисту електронної пошти *ProofPoint*, яка використовує заголовки електронних листів для детектування шкідливого вмісту. Спочатку надсилається велика кількість електронних листів та збираються оцінки моделі *ML Proofpoint*. Визначається, яка змінна в оцінці відповідає за безпеку електронної пошти. Використовуючи ці оцінки, дослідники відтворили режим машинного навчання, побудувавши ФЕМ. Висновки, отримані внаслідок використання офлайн-моделі, дозволили дослідникам створювати шкідливі

електронні листи, які змогли обійти системи захисту електронної пошти *ProofPoint*. Порядок дій дослідників зображено на рисунку 7.

Рис. 6. Схема обходу моделі виявлення ШПЗ

Рис. 7. Схема ухилення від детектування шкідливого вмісту в електронних листах

Висновки

У роботі досліджено *ML*-моделі виявлення вторгнень, аномалій, фішингових атак, шкідливого програмного забезпечення та ботнетів, які реалізовані на основі дерева рішень, методу опорно-векторних машин, *k*-найближчих сусідів, наївного баєсового класифікатора, логістичної регресії, а також штучних нейронних мереж. Розглянуто існуючі *AML*-моделі та проведено аналіз прикладів їхнього застосування для здійснення конкурентних атак ухилення, отруєння, функціонального вилучення на системи кіберзахисту на основі *ML*-моделей.

Дослідження показали, що навіть не маючи доступу до алгоритмів роботи моделей машинного навчання, можливо реалізувати обхід системи кіберзахисту. Наприклад, дослідницька група *«Palo Alto Networks Security AI»* досягла зниження ефективності виявлених ботнетів моделлю глибокого навчання на 25 % та обійшла згорткову нейронну мережу виявлення ШПЗ в *HTTP*-трафіку у 80 % випадків. Шляхом ефективної реалізації моделі ухилення від виявлення ШПЗ антивірусним програмним засобом «Cylance» дослідникам компанії *«SkylightCyber»* вдалося обійти захист в 88,4 % випадків. Здійснивши атаку на відому платформу обміну вірусами *«VirusTotal»*, зловмисникам вдалось отруїти навчальний набір даних *ML*-моделі, що призвело до хибного детектування програм-вимагачів.

Отже, для забезпечення безпеки мереж і послуг засобами кіберзахисту зі штучним інтелектом необхідно враховувати необхідність протидії конкурентним атакам, а саме:

- забезпечувати збір та агрегацію навчальних даних кожною системою окремо, а не отримувати їх із загально-доступних джерел;
- здійснювати оптимізацію вмісту журналів подій, з урахуванням можливості використання інформації, що знаходиться в них, для створення конкурентних атак;
- у випадку розгортання систем кіберзахисту на об'єктах критичної інфраструктури використовувати спеціально розроблені *ML*-моделі, яких немає в загальному доступі, що ускладнить можливість створення їх ФЕМ;
- забезпечити захист навчальних даних та алгоритмів функціонування *ML*-моделей.

Напрямком подальших досліджень є аналіз існуючих методів підвищення стійкості систем кіберзахисту зі штучним інтелектом до впливу конкурентних атак.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Al-Omari M., Rawashdeh M., Qutaishat F., Alshira'H M., Ababneh N. An intelligent tree-based intrusion detection model for cyber security // Journal of Network and Systems Management. 2021. Vol. 29 (2). P. 1–18. DOI: 10.1007/s10922-021-09591-y.

2. Xin, Yang & Kong, Lingshuang & Liu, Zhi & Chen, Yuling & Li, Yanmiao & Zhu, Hongliang & Mingcheng, Gao & Hou, Haixia & Wang, Chunhua. Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access. 2018. P. 1-1. URL: https://www.researchgate.net/publication/325159145_Machine_ Learning_and_Deep_Learning_Methods_for_Cybersecurity.

3. Heinrich, Kai & Graf, Johannes & Chen, Ji & Laurisch, Jakob & Zschech, Patrick. Fool me Once, Shame on you, Fool me Twice, Shame on me: A Taxonomy of Attack and Defense Patterns for AI Security. 2020. URL: https://www.researchgate.net/publication/341180631_Fool_me_Once_Shame_on_you_Fool_me_Twice_Shame_on_me_A_Taxonomy_of_Attack_and_Defense_Patterns_for_AI_Security.

4. NIST AI 100-2e2023 ipd. Adversarial Machine Learning. A Taxonomy and Terminology of Attacks and Mitigations. URL: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf.

5. Kianpour, Mazaher & Wen, Shao-Fang. Timing Attacks on Machine Learning: State of the Art. 2020. URL: https://www.researchgate.net/publication/335382520_Timing_Attacks_on_Machine_Learning_State_of_the_Art.

6. P. I. Radoglou-Grammatikis and P. G. Sarigiannidis. An Anomaly-Based Intrusion Detection System for the Smart Grid Based on CART Decision Tree // 2018 Global Information Infrastructure and Networking Symposium (GIIS), Thessaloniki, Greece, 2018. P. 1–5. DOI: 10.1109/GIIS.2018.8635743.

7. Moorthy R. S., Pabitha P. Optimal detection of phising attack using SCA based K-NN // Procedia Computer Science. 2020. Vol. 171. P. 1716–1725.

8. S.-Y. Lee, S.-r. Wi, E. Seo, J.-K. Jung and T.-M. Chungю ProFiOt: Abnormal Behavior Profiling (ABP) of IoT devices based on a machine learning approach // 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, VIC, Australia, 2017. P. 1–6. DOI: 10.1109/ATNAC.2017.8215434.

9. Swarnkar M., Hubballi N. OCPAD: One class Naive Bayes classifier for payload based anomaly detection // Expert Syst. Appl. Oct. 2016. Vol. 64. P. 330–339.

10. R. Bapat et al. Identifying malicious botnet traffic using logistic regression // Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2018. P. 266–271. DOI: 10.1109/SIEDS.2018.8374749.

11. Katja Auernhammer, Ramin Tavakoli Kolagari, and Markus Zoppelt. Attacks on Machine Learning: Lurking Danger for Accountability. AAAI 2019. URL: https://www.researchgate.net/publication/330737530_Attacks_on_Machine_Learning_Lurking_Danger_for_Accountability.

12. S. Herpig. Securing artificial intelligence – Part 1: The attack surface of machine learning and its implications. Think Tank at the Intersection of Technology and Society, Stiftung Neue Verantwortung, Berlin, Oct. 2019. [Online]. URL: https://www.stiftung-nv.de/sites/default/files/securing_artificial_intelligence.pdf.

13. ATLAS Matrix. URL: https://atlas.mitre.org/matrices/ATLAS/.

14. Adversarial Machine Learning. URL: https://atlas.mitre.org/resources/adversarial-ml-101/.

15. Goodfellow, Ian & Shlens, Jonathon & Szegedy, Christian. (2014). Explaining and Harnessing Adversarial Examples. arXiv 1412.6572. URL: https://www.researchgate.net/publication/269935591_Explaining_and_Harnessing_Adversarial_Examples.

16. Peter Lee. Learning from Tay's introduction // Microsoft. 2016. URL: https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/.

17. Le, Hung & Pham, Quang & Sahoo, Doyen & Hoi, Steven. URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection (2018). URL: https://www.researchgate.net/publication/323118482_URLNet_Learning_a_URL_Representation_with_Deep_Learning_for_Malicious_URL_Detection.

18. Evasion of Deep Learning Detector for Malware C&C Traffic. Actor: Palo Alto Networks AI Research Team. Incident Date: 2020.

19. Botnet Domain Generation Algorithm (DGA) Detection Evasion. Actor: Palo Alto Networks AI Research Team. Incident Date: 2020. URL: https://atlas.mitre.org/studies/AML.CS0001.

20. B. Yu, J. Pan, J. Hu, A. Nascimento and M. De Cock. Character Level based Detection of DGA Domain Names, 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1–8. URL: https://ieeexplore.ieee.org/document/8489147.

21. VirusTotal Poisoning. Incident Date: 2020. URL: https://atlas.mitre.org/studies/AML.CS0002.

22. Metame: metamorphic code engine for arbitrary executables. URL: https://github.com/a0rtega/metame.

23. Bypassing Cylance's AI Malware Detection. Actor: Skylight Cyber. 2019. URL: https://skylightcyber.com/2019/07/18/cylance-i-kill-you/.

24. CVE-2019-20634 Detail. National Vulnerability Database. 2022. URL: https://nvd.nist.gov/vuln/detail/CVE-2019-20634.