СТАТТІ ТА ОГЛЯДИ

doi: https://doi.org/10.15407/visn2023.02.012

ЛЕТИЧЕВСЬКИЙ Олександр Олександрович — доктор фізико-математичних наук, завідувач відділу теорії цифрових автоматів Інституту кібернетики імені В.М. Глушкова НАН України

СУЧАСНІ НАУКОВІ ПРОБЛЕМИ КІБЕРБЕЗПЕКИ

У статті наведено огляд сучасних проблем кібербезпеки та проаналізовано роль наукових досліджень у їх вирішенні. Зокрема, виокремлено два види досліджень — із застосуванням алгебраїчного підходу та з використанням нейронних мереж, що належать до методів штучного інтелекту. Алгебраїчні методи ґрунтуються на використанні автоматичного доведення теорем та програм-розв'язувачів і спрямовані на вирішення двох основних проблем кібербезпеки: перша — виявлення вразливостей у програмних і апаратних системах та оцінка їх стійкості до вторгнень; друга — виявлення вторгнень зловмисників у реальному часі. Наведено результати досліджень, які сприяють створенню надійного захисту систем від кібератак, зокрема систем об'єктів критичної інфраструктури, що на сьогодні є особливо актуальним завданням.

Ключові слова: кібербезпека, кібератака, алгебраїчне моделювання, вразливості коду, інсерційне моделювання, формальна верифікація, нечітке тестування, нейронні мережі, машинне навчання, алгебра поведінок.

Результати війни в кіберпросторі є визначальними для загального розвитку бойових дій у сучасних протистояннях. Кібернапад та кібервторгнення можуть завдати величезних збитків або спричинити значні руйнування критичної інформаційної інфраструктури на будь-якому рівні. Це стосується насамперед кібератак на об'єкти енергетичної, транспортної та військової інфраструктури, під час яких виводяться з ладу об'єкти управління постачанням, керування логістикою тощо.

Такі атаки ми спостерігаємо протягом усього періоду протистояння з РФ, починаючи з 2014 р. Всі пам'ятають, як наприкінці 2015 р. сталися кібератаки на інфраструктурні об'єкти енергетичної галузі на Прикарпатті та у Львівській області, внаслідок яких цілі регіони залишилися без електропостачання і знадобився певний час для відновлення мережі. В ніч широкомасштабного воєнного вторгнення РФ на територію України 24 лютого 2022 р. було здійснено масовану атаку на українські мережі, але завдяки завчасно вжитим заходам з кіберзахисту її наслідки вдалося мінімізувати. Останнім часом відбувалися тривалі атаки на банківські установи, які все ж завдали шкоди, однак унаслідок швидкого реагування систем кіберзахисту ці

збитки виявилися не такими відчутними, як могло б бути. Спостерігалися також атаки на сайти державних установ та органів центральної влади, в результаті чого зловмисники розміщували на них провокаційні оголошення або виводили їх з ладу за допомогою так званих DDoS-атак.

Основні засоби кібернападу пов'язані з використанням шкідливого коду та спробами вторгнення за допомогою вразливостей систем. Шкідливий код найчастіше потрапляє в систему внаслідок порушення користувачами кібергігієни — перехід на небезпечні сайти, відкриття вкладень у підозрілих листах з електронної пошти, а тому ступінь успішності вторгнення визначається якістю системи захисту.

У кібербезпеці можна виділити п'ять ключових активностей для протистояння вторгненню:

1) виявлення атаки, тобто фіксація факту аномальної роботи мережі;

2) ідентифікація атаки, що полягає в точній класифікації її різновиду;

3) захист від атаки, що охоплює послідовність кроків, спрямованих на протидію вторгненню;

4) відповідь у разі ідентифікації нападника;

5) ліквідація наслідків атаки та відновлення інформації з використанням алгоритмів відновлення та збереження даних.

Кожна з цих активностей є окремим напрямом в інженерних та прикладних науках. Крім того, у процесі підготовки до можливих атак слід забезпечити відповіді на два ключових запитання:

1) наскільки система стійка до вторгнень і здатна до протидії кібератакам?

2) яка якість системи відносно наявності вразливостей у програмному чи апаратному забезпеченні?

Якщо перше питання стосується ефективності системи виявлення вторгнень у процесі взаємодії з атаками, то друге — якості тестування та верифікації системи на етапі її створення.

У сучасних технологіях кібербезпеки активно використовують як інженерні методи, так і останні досягнення науки, такі як методи штучного інтелекту та формальні алгебраїчні методи. Понад 5 років тому в Інституті кібернетики імені В.М. Глушкова НАН України було започатковано дослідження, спрямовані на підвищення ефективності засобів кіберзахисту з використанням новітніх наукових здобутків у цих напрямах. Зокрема, активно впроваджується алгебраїчний підхід, що відкриває можливості для найбільш точного аналізу моделей програмних та апаратних систем з метою визначення стійкості до вторгнень та відсутності вразливої поведінки. Іншим підходом є використання нейронних мереж глибокого навчання, що вбудовуються в систему виявлення вторгнень (Intrusion Detection System — IDS) [1].

Сучасні системи виявлення вторгнень. Вторгнення можна визначити як будь-який вид несанкціонованої діяльності, яка завдає шкоди інформаційній системі. Сучасні системи виявлення вторгнень забезпечують істотне поліпшення функцій захисту порівняно з попередніми засобами кібербезпеки, такими як мережевий екран, віртуальна приватна мережа та шифрування сповіщень. Такі системи виконують дві головних функції. По-перше, система виявляє небажану поведінку у вигляді аномалії, навіть якщо це може не бути справжнім вторгненням (хибне виявлення). По-друге, система збирає дані, аналізує дії в мережевих протоколах та порівнює їх з так званими сигнатурами, що містять дані про можливі атаки. Відповідно до цих двох функцій розрізняють два основні типи систем виявлення, хоча насправді є багато їх різновидів, які поєднують у собі ці дві функції.

Сигнатури відомих атак існують у базі даних системи виявлення в різних специфікаціях, наприклад у формі правил над параметрами протоколу у вигляді if-then-else. Якщо правило, яке міститься в базі даних сигнатур і кваліфікується як певний вид вторгнення, виконується для параметрів протоколу, то спрацьовує сигнал тривоги.

Є певні труднощі з виявленням вторгнень, що проявляються впродовж деякого проміжку

часу і містять ознаки вторгнення в різних пакетах трафіку протоколу. Для подолання цих ускладнень деякі системи використовують представлення сигнатур скінченних автоматів, зокрема як це реалізовано в розробках Мічиганського університету [2]. У Національному університеті Тайваню в системах застосовують шаблони мовних рядків чи семантичні умови [3]. Однак вторгнення, розтягнуті в часі, виявляються не завжди.

Вважається, що порівняння з сигнатурами для таких систем виявлення вторгнень є досить дієвим методом, який дає непогані результати у виявленні вже відомих атак, але в разі атак нульового дня, тобто невідомих раніше, вони безсилі.

Системи на основі виявлення аномалій можуть зафіксувати такі невідомі раніше вторгнення, як відхилення від нормальної поведінки мережевої активності. Серед цих систем виділяють кілька різновидів.

Системи виявлення на основі статистики створюють модель розподілу подій для нормальної поведінки, потім виявляють події з низькою ймовірністю та позначають їх як потенційні вторгнення.

Системи, засновані на знаннях, використовують факти про нормальну діяльність мережевого протоколу і будь-яке відхилення класифікують як вторгнення. Недоліком цього методу є те, що зібрати всі факти про нормальну роботу системи дуже складно, навіть з використанням формалізації роботи протоколу за допомогою формальних конструкцій. Такими конструкціями можуть бути скінченні автомати, що розглянуто в роботах британського Університету Лестера [4] та Університету Тулузи [5].

Більш сучасні системи на основі виявлення аномалій найчастіше використовують машинне навчання. Вони демонструють кращу точність як на відомих атаках, так і на вторгненнях нульового дня. Крім того, такі системи в разі тренування їх на правильних даних можуть класифікувати відомі атаки, хоча при цьому виникають інші проблеми.

Машинне навчання — це процес отримання знань з великої кількості даних з метою розпізнавання чи прогнозування поведінки. Знання формуються у вигляді моделі класифікації, що забезпечується певним алгоритмом генерації. Для побудови моделей класифікації поведінки мережі використовують алгоритми кластеризації, генерації нейронних мереж, генетичні алгоритми, дерева рішень та метод k-найближчих сусідів.

На сьогодні нейронні мережі є основною моделлю в системах виявлення вторгнень. Використання нейронних мереж детальніше розглянемо в наступних розділах.

Виявлення вразливостей у програмних та апаратних системах. Вразливості програмного забезпечення є основною ціллю атак, які можуть завдати шкоди роботі та репутації мільйонів систем у всьому світі, а також призвести до величезних фінансових втрат. Тому виявлення вразливостей як у програмному, так і в апаратному забезпеченні є одним з головних завдань кібербезпеки.

Засоби виявлення вразливостей уже давно застосовують у системах розроблення програм, а також як окремі системи виявлення. Виявлення вразливостей розглядають як на рівні початкового коду в мові програмування високого рівня, так і на рівні бінарного коду. Методам виявлення вразливостей присвячено досить велику кількість публікацій, зокрема можна згадати огляди дослідників Університету Аделаїди (Австралія) [6] або Уханського університету (Китай) [7].

Системи розроблення програм пропонують виявлення вразливостей на основі помилкових фрагментів коду, що дає можливість побудувати так званий експлойт (exploit). Експлойт —це сценарій поведінки та відповідні дані, використовуючи які зловмисник може здійснити вторгнення з метою знищення системи, порушення ідентифікації або перехоплення керування системою.

Основні недоліки використання фрагментів програмного коду полягають у тому, що цей метод не гарантує відсутності інших вразливостей, тобто вразливість може існувати, наприклад, у бібліотеках, які використовує програма, і аналіз на рівні початкового коду її не

виявить. Зокрема, це стосується некоректного використання бібліотек. Крім того, виявлення вразливих фрагментів коду може бути хибним, тобто знайдена вразливість ніколи не спрацює при виконанні програм.

Іншим, більш розвиненим засобом представлення вразливостей є їх формальні шаблони. При цьому застосовують методи моделювання коду, хоча й вважається, що ці методи дають досить низьке покриття.

Обидва підходи використовують системи виявлення вразливостей у бінарному коді. У цьому разі постає вже проблема адекватного представлення вразливостей на рівні бінарного коду, що виходить з мови програмування, якою написано програму.

У разі виявлення вразливостей важливими є подальші дії. Якщо на рівні програмування мовами високого рівня пропонується замінювати помилкові фрагменти більш безпечними, то на рівні бінарного коду розглядається застосування технології автоматичного виправлення (patching). З одного боку, автоматичне виправлення може призвести до непередбачуваної поведінки програми, але з іншого — воно може бути коректним, якщо виправлення еквівалентне, тобто не змінює поведінку програми, що має бути перевірено формальними методами. Так, на змаганнях з кібербезпеки, організованих Агентством передових оборонних дослідницьких проєктів США (DARPA) [8], такі виправлення проводилися в контексті символьних обчислень.

Зі зростанням популярності програмованих плат пошук вразливостей в апаратних специфікаціях набув нового сенсу. Такі компанії, як Cadence, Xilinx, Synopsis, значну увагу приділяли моделюванню специфікацій для мов електронного дизайну, зокрема VHDL та System Verilog. При цьому перевірялися такі властивості, як відсутність перегонів сигналів, переповнення буферу тощо.

При верифікації мов електронного дизайну виникали проблеми, пов'язані з відсутністю вразливостей після перенесення коду на нижчий рівень, яке виконує відповідний компілятор. Однак при трансляції коду в набір команд інтегральних плат та оптимізації деякі вразливості могли з'явитися знову. Тому проблема еквівалентності бінарного та початкового коду є важливою для виробників апаратного забезпечення.

Усі розглянуті методи добре працюють для виявлення вразливостей, які відомі і занесені у відповідні світові бази даних вразливостей CVE/CWE. Для пошуку невідомих вразливостей використовують техніку, яку називають *нечітке тестування*. Вперше її було застосовано в проєкті SAGE [9], започаткованому компанією Microsoft для виявлення помилок в операційній системі Windows. У цьому проєкті генератор створював вхідні дані, які вважалися неочікуваними, наприклад максимально великі числа чи рядки символів дуже великої довжини. Дані подавали на вхід програми, що тестувалася, та очікували результат, що мав перевірити реакцію системи на аномальні вхідні дані. Це тестування проводили цілодобово, і зрештою вдалося виявити кілька десятків невідомих вразливостей, які було занесено в базу даних вразливостей CVE. Інша система нечіткого тестування AFL [10], розроблена компанією Google, крім того, що генерувала неочікувані дані, ще й стежила за покриттям рядків коду і, відповідно, створювала тести.

Алгебраїчний підхід у завданнях кіберзахисту. Більш надійний спосіб виявлення вразливостей стало можливим реалізувати завдяки використанню формальних методів. Уперше підтвердження цьому було отримано під час змагання з виявлення вразливостей та протидії кібератакам, організованого у 2016 р. агентством DARPA [8], де перші три медалісти використовували виключно алгебраїчний підхід.

Потужна алгебраїчна школа В.М. Глушкова, яка продовжує розвиватися в Інституті кібернетики, та здобутки в розвитку методів автоматичного доведення теорем створили передумови для появи алгебраїчного підходу у вирішенні практичних завдань. Так, було створено систему алгебраїчного програмування [11] та теорію взаємодії агентів і середовищ [12] (спільно з британським науковцем Д. Гільбертом). Результатом подальшого розвитку цих

досліджень було створення теорії інсерційного моделювання [13] та алгебри поведінок [14], які стали узагальненням теорії автоматів і теорії транзиційних систем. Алгебра поведінок виявилася досить розвиненим математичним апаратом для формалізації як математичних, так і природних об'єктів, на моделях яких було розроблено велику кількість формальних методів.

З точки зору кібербезпеки алгебра поведінок дала можливість описувати поведінку агентів у мережевому середовищі та вивчати їхні властивості. Для цього було розроблено технологію алгебраїчних сигнатур на рівні бінарного коду програмної системи.

Алгебраїчна сигнатура — це поведінка атакера чи вразлива поведінка програми, що представлена за допомогою формули алгебри поведінок. Відмінність такого представлення від традиційних сигнатур вірусів полягає у формалізації поведінки на більш високому рівні абстракції. В сигнатурах вірусів містилися конкретні значення байтів зловмисного коду, за якими віруси розпізнавалися антивірусними програмами.

Нижче наведено приклад фрагменту такої сигнатури:

B = X; Y; a3,Y = a1.Z;a1;(Y + a3.Y),

Z = G;a2;F,

a1 = (id == rdtscp) ->1,

a2 = (id == mov && dest ==
== Mem(CASH)) ->1,

a3 = (id == mov && dest ==
==ADDR(EAX)) -> 1.

Зазначені формули представляють рівняння алгебри поведінок. Цю поведінку можна визначити так: «Один із процесів вимірює в циклі час доступу до даних у кеш-пам'яті та використовує непрямий доступ до комірок основної пам'яті». Така поведінка характерна для атаки Meltdown.

Один із різновидів вразливості «пошкодження стеку» можна представити у вигляді поведінкових рівнянь:

StackVulnerability = AllocateStack; Write-
Stack,

WriteStack = (mov(l, Memory(s),
GeneralRegister).Delta + !mov(l,
Memory(s),GeneralRegister)).WriteStack,

AllocateStack = push(1, ebp).mov(2, ebp,
esp).sub(3, esp, Numeric),

mov(l, Memory(s), GeneralRegister) =
=Dirty(GeneralRegister) & (s <= BP) -> 1,

mov(m, Memory(s), GeneralRegister) =
=!(m = l) & Dirty(GeneralRegister) ->
Dirty(s),

mov(n, Memory(s), GeneralRegister) =
=!(n = l) & (Dirty(s) &
!Dirty(GeneralRegister)) -> !Dirty(s)

mov(m, GeneralRegister, Memory(s)) =

=!(m = l) & Dirty(s) ->
Dirty(GeneralRegister),

mov(n, GeneralRegister, Memory(s))=
=!(n = l) & (!Dirty(s) &
Dirty(GeneralRegister)) ->
!Dirty(GeneralRegister).

Формулою поведінкових рівнянь можна описувати й інші вразливості, які зафіксовані у світових базах даних вразливостей CVE/CWE.

База даних CVE (Common Vulnerabilities and Exposures) містить усі відомі на сьогодні вразливості і точну інформацію про кожну з них: продукт, у якому її виявлено, ступінь впливу на функціонування програми, тип вразливості та інші дані.

База даних CWE (Common Weakness Enumeration) містить категорії вразливостей. Якщо CVE — це список відомих на сьогодні проблем, то CWE класифікує ці проблеми і дає більш узагальнений їх опис.

Ці алгебраїчні сигнатури вразливостей є описом верхнього рівня і покривають множини сценаріїв за різних вхідних даних програм. Вразливості можна розпізнати відповідними методами виявлення, створеними в рамках алгебри поведінок. Аналогічно можуть бути задані формули атак, що покривають множину поведінок зловмисної програми, які також можна розпізнати в підозрілому коді.

Для виявлення поведінок в Інституті кібернетики імені В.М. Глушкова НАН України

спільно з Херсонським державним університетом було створено програму *алгебраїчного зіставлення* [15]. Щоб здійснити таке розпізнання у програмному або апаратному забезпеченні, необхідно було провести сканування його бінарного коду чи схеми. Далі сканований бінарний код транслюється в модель алгебри поведінок, яка зіставляється з наявними алгебраїчними сигнатурами вразливих та зловмисних поведінок.

Програма алгебраїчного зіставлення використовує технологію автоматичного доведення теорем, символьне моделювання та методи генерації інваріантів. При цьому використовують сучасні машини доведення, такі як Microsoft Z3 [16] та системи-розв'язувачі.

Головною властивістю алгебраїчного підходу, на відміну від тестування та інших технологій виявлення вразливостей, є те, що метод надає *доказове підтвердження* відсутності вразливостей, а в разі їх наявності представляє сценарій, який приводить до спрацювання вразливості. Такий сценарій називають експлойтом (exploit).

З іншого боку, метод алгебраїчного зіставлення використовує обчислення високої складності та алгоритми, що є нерозв'язними в рамках деяких теорій. Як наслідок, може виникнути комбінаторний вибух, що призведе до неможливості довести відсутність вразливості.

Сучасна наука має досить велику кількість методів, спрямованих на боротьбу з комбінаторним вибухом, наприклад апроксимаційний підхід. Оскільки критичним для застосування алгебраїчного зіставлення є час, використання цього методу найбільш ефективне в період розроблення системи, на етапі підготовки її до запуску, а також під час аналізу підозрілих процесів у мережі чи на настільному комп'ютері.

В іншому разі метод може працювати як своєрідний алгебраїчний антивірус. У цьому випадку код підозрілої програми сканується, транслюється в рівняння алгебри поведінок та аналізується алгебраїчним зіставленням з базою даних алгебраїчних сигнатур. При цьому, що важливо, немає потреби запускати підозрілу програму, на відміну від так званих «пісочниць» [17], де підозрілі програми запускають в ізольованому середовищі.

Ще однією перевагою алгебраїчних сигнатур є те, що вони спроможні частково протистояти такому явищу, як змінюваність атак. Оскільки атака описана на верхньому рівні, вона покриває множину дій хакера. Тому, щоб обійти захист алгебраїчного антивірусу, потрібно створити принципово нову атаку, яка не збігається з уже відомими алгебраїчними сигнатурами.

З метою підвищення ефективності аналізу вразливостей у великих системах в Інституті кібернетики ведуться роботи з розгортання паралельної версії системи алгебраїчного зіставлення на суперкомп'ютері СКІТ-4. Це дає змогу в сотні разів прискорити виконання аналізу вразливостей для великих систем, а також для великої кількості аналізованих вразливостей. Для доступу в таку систему достатньо на місці просканувати програмний код спеціальною програмою, яка транслює його в модель алгебри поведінок, і надіслати для аналізу на СКІТ-4. Результатом аналізу буде список вразливостей, що знайдені в системі, та сценарії вторгнень, можливих у рамках цієї системи. Цей сервіс планується використовувати для кіберзахисту об'єктів критичної інфраструктури.

Слід згадати й інші роботи, що проводяться в Інституті кібернетики з використанням алгебраїчної технології. Серед них — дослідження протоколів Інтернету речей, зокрема «розумного будинку», на вразливості [18]. Одна з поширених вразливостей — це перехоплення контролю за сповіщенням. Завдяки вразливості у запитах HTTP-протоколу зловмисник може переспрямовувати листи від визначених відправників через свою адресу. Для встановлення таких вразливостей необхідно формалізувати протокол та визначити властивості безпеки, у цьому випадку — конфіденційності листування. Формальними методами в алгебрі поведінок, зокрема символьним моделюванням, можна визначити порушення властивості безпеки та згенерувати приклад, що веде до цього порушення.

З використанням рівнянь алгебри поведінок проводяться роботи зі створення системи нечіткого тестування [19]. У цій системі, крім керування покриттям бінарного коду, аналізуються потенційні місця, де можуть бути вразливості, та генеруються відповідні тести, завдяки чому цей метод є більш керованим, ніж відома система AFL.

Нейронні мережі та виявлення вторгнень. Алгебраїчні методи застосовують для перевірки властивостей пошуку вразливостей системи, яка може зазнати вторгнення. При цьому складність обчислень може бути досить високою і перевірка потребуватиме багато часу. Однак при виявленні атак у процесі функціонування системи час є критичним, і алгебраїчний підхід може виявитися неефективним.

Для швидкого виявлення атаки в мережевому середовищі використовують нейронні мережі глибокого навчання (deep neuron networks), які здатні класифікувати поведінку мережевого протоколу при вторгненні як аномальну. Більш розвинені нейронні системи визначають тип атаки згідно з її моделлю класифікації. Як уже було заначено вище, нейронні мережі використовують у системах виявлення вторгнень (IDS).

Нейроні системи будують за допомогою машинного навчання або тренування на певних наборах даних, які збирають у процесі спостереження поведінки мережевих протоколів. Найпростіший і найшвидший спосіб виявлення атак у реальному часі — це визначення аномальної поведінки, яка не відповідає нормальним діям протоколу. Однак при цьому можливі хибні виявлення, оскільки відхилення від нормальної поведінки можуть виникати не лише через вторгнення в систему, а й унаслідок неправильного користування ресурсами, помилок користувачів або програм, що функціонують у середовищі. Тому перевагу віддають системам, які спроможні класифікувати причини аномалії. Проте такі системи не зможуть визначити атаку, для якої не було створено тренувального набору.

Тому значну увагу приділяють наборам даних, призначених для тренування. На сьогодні є кілька відкритих наборів даних, що містять поведінку мережевих протоколів, однак з їх використанням виникають певні труднощі. Однією з основних проблем є те, що кількість даних недостатня для точної класифікації певних атак. Дані можуть бути надлишковими або зашумленими. При цьому може виникати дисбаланс у розподілі класів моделі. Оскільки критичним є час класифікації поведінки, розглядалися нейронні мережі глибокого навчання з мінімальною кількістю шарів для скорочення обчислень. Також для підвищення ефективності використовували інші типи мереж: згорткові та рекурентні нейронні мережі.

Задача генерації моделі класифікації є досить нетривіальною для практичного застосування, оскільки відсутність наборів тренувальних даних в умовах постійних швидких змін використовуваних хакерами технологій є вагомою проблемою. Сучасні хакери розуміють природу моделі класифікації відхилень і намагаються замаскувати свої дії під нормальну поведінку в протоколі. У зв'язку з цим постає проблема подолання такого ухиляння хакерів.

Як в ухилянні від виявлення антивірусними програмами просто за допомогою перестановки кількох байтів і уникання в такий спосіб зіставлення з конкретними сигнатурами вірусів, так і у протидії системам виявлення вторгнень, хакер використовує у своїх діях еквівалентні перетворення, щоб вони сприймалися як нормальна поведінка. Такими перетвореннями можуть бути перестановка компонент проведення атаки, вставлення додаткових символів розділення, використання альтернативних кодувань, нестандартних портів та безліч інших хитрощів, які унеможливлюють класифікацію поведінки мережевого протоколу як аномальної. На сьогодні ця проблема є досить мало дослідженою.

Одним зі шляхів вирішення зазначеної проблеми є створення набору з розширених даних, який охоплює можливі відомі перетворення, що використовують хакери. Доповнення таким набором приводить до більшої надійності системи та забезпечує можливість виявляти вторгнення з використанням хакерського ухиляння.

Інститут кібернетики співпрацює з единбурзьким Університетом Геріот-Ватт у започаткованому проєкті верифікації нейронних мереж з метою визначення їх ефективності та стійкості до ухиляння.

Натреновану нейронну мережу перевіряють на можливість неправильної класифікації, що виражається формулою алгебри поведінок. При цьому застосовують алгебраїчне моделювання нейронної мережі для виявлення обмежень тренувальних наборів, за яких можлива неправильна класифікація. Планується тестове застосування більш надійних нейронних мереж такого типу насамперед на протоколах Інтернету речей та блокчейнового мережевого середовища.

Ще одним завданням з використанням нейронних мереж є виявлення так званих ботнетів, які створюють загрозу DDoS-атак.

Атака DDoS є одним з найпоширеніших методів виведення з ладу вебсайтів та серверів. Атаці передує період встановлення на комп'ютерних системах зловмисного коду, який активізується з метою одночасного запиту до інтерфейсу системи великої кількості користувачів, що створює надмірне навантаження на систему, після чого вона виходить з ладу.

За допомогою аналізу мережевого протоколу нейронна мережа класифікує особливу поведінку, притаманну ботнетам, які активізуються або встановлюються. Своєчасне виявлення відхилень у поведінці протоколу від заражених комп'ютерних ресурсів дасть змогу нейтралізувати DDoS-атаку. В Інституті кібернетики з цього напряму проводять відповідні дослідження.

Висновки. Отже, використання методів штучного інтелекту, таких як машинне навчання та алгебраїчні дедуктивні методи, є більш ефективним у вирішенні проблем кібербезпеки, ніж інженерні рішення, які ґрунтуються на вірусних і конкретних поведінкових сигнатурах та карантинних «пісочницях».

Стрімке зростання якості і кількості машин автоматичного доведення та системрозв'язувачів разом зі збільшенням числа різновидів нейронних мереж дає можливість створити ефективні системи виявлення вторгнень та системи аналізу стійкості програмного й апаратного забезпечення. Підвищення швидкодії комп'ютерних систем та використання паралельних обчислень може пришвидшити наступний етап створення систем виявлення вторгнень у реальному часі саме на основі алгебраїчних методів, які відкривають більші можливості як для ширшої класифікації, так і для вищої точності виявлень. Одним з проміжних етапів може бути комбіноване використання нейронних мереж та алгебраїчного зіставлення.

Поповнення баз даних алгебраїчних сигнатур є одним із завдань у комунікації різних груп кібербезпеки. Обмін тренувальними наборами даних разом із доповненням новими випадками вторгнень створить постійну складову, яка підвищуватиме спроможність та надійність нейронних мереж у кібербезпеці.

REFERENCES

1. Khraisat A., Gonda I., Vamplew P., Kamruzzaman J. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecur.* 2019. **2**: 20. https://doi.org/10.1186/s42400-019-0038-7
2. Meiners C.R., Patel J., Norige E., Torng E., Liu A.X. Fast regular expression matching using small TCAMs for network intrusion detection and prevention systems. In: *USENIX Security'10*: Proc. 19th USENIX Conf. on Security. Washington, DC, 2010.
3. Lin C., Lin Y.-D., Lai Y.-C. A hybrid algorithm of backward hashing and automaton tracking for virus scanning. *IEEE Trans. Comput.* 2011. **60**(4): 594—601. https://doi.org/10.1109/TC.2010.95
4. Walkinshaw N., Taylor R., Derrick J. Inferring extended finite state machine models from software executions. *Empirical Software Engineering.* 2016. **21**(3): 811—853. https://doi.org/10.1007/s10664-015-9367-7
5. Studnia I., Alata E., Nicomette V., Kaâniche M., Laarouchi Y. A language-based intrusion detection approach for automotive embedded networks. *Int. J. Embed Syst.* 2018. **10**(1): 1—12. https://doi.org/10.1504/IJES.2018.089430

6. Le T.H.M., Chen H., Ali Babar M. A Survey on Data-driven Software Vulnerability Assessment and Prioritization. *ACM Computing Surveys*. 2023. **55**(5): 1—39. https://doi.org/10.1145/3529757
7. Shen Z., Chen S. A Survey of Automatic Software Vulnerability Detection, Program Repair, and Defect Prediction Techniques. *Security and Communication Networks*. 2020. **2020**: 8858010. https://doi.org/10.1155/2020/8858010
8. Cyber Grand Challenge. DARPA. https://www.darpa.mil/program/cyber-grand-challenge
9. Godefroid P., Levin M.Y., Molnar D. SAGE: Whitebox Fuzzing for Security Testing. *Queue*. 2012. **10**(1): 20—27. https://doi.org/10.1145/2090147.2094081
10. American Fuzzy Lop. https://lcamtuf.coredump.cx/afl/
11. Kapitonova J., Letichevsky A. Algebraic programming in the APS system. In: *ISSAC 90*: Proc. Int. Symp. on Symbolic and Algebraic Computation. ACM, New York, 1990. P. 68—75. https://doi.org/10.1145/96877.96896
12. Gilbert D., Letichevsky A. A model for interaction of agents and environments. In: Bert D., Choppy C. (eds). *Recent Trends in Algebraic Development Techniques*. LNCS 1827. Cham, Switzerland: Springer-Verlag, 1999. P. 311—328. https://doi.org/10.1007/978-3-540-44616-3_18
13. Letichevsky A., Letychevskyi O., Peschanenko V., Weigert T. Insertion modeling and symbolic verification of large systems. In: Fischer J., Scheidgen M., Schieferdecker I., Reed R. (eds). *SDL 2015: Model-Driven Engineering for Smart Cities*. Cham, Switzerland: Springer International Publishing, 2015. P. 3—18. https://doi.org/10.1007/978-3-319-24912-4_1
14. Letichevsky A. Algebra of behavior transformations and its applications. In: Kudryavtsev V.B., Rosenberg I.G. (eds). *Structural Theory of Automata, Semigroups, and Universal Algebra*. NATO Science Series II. Mathematics, Physics and Chemistry. Vol. 207. Springer, 2005. P. 241—272. https://doi.org/10.1007/1-4020-3817-8_10
15. Letychevskyi O. Two-level algebraic method for detection of vulnerabilities in binary code. In: *10th IEEE Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 2019. https://doi.org/10.1109/IDAACS.2019.8924255
16. Z3 decision procedure. https://github.com/Z3Prover/z3
17. Pulapaka H. Windows sandbox. *Windows OS Platform Blog*. https://techcommunity.microsoft.com/t5/windows-kernel-internals-blog/windows-sandbox/ba-p/301849
18. Horbatyuk V.O., Horbatyuk S.O. Methods for Checking the Resistance to http Attacks on a Smart Home by Algebraic Comparison. *Control Systems and Computers*. 2022. (4):13—23. https://doi.org/10.15407/csc.2022.04.013
19. Letychevskyi O.O., Peschanenko V.S., Hryniuk Y.V. Fuzz Testing Technique and its Use in Cybersecurity Tasks. *Cybernetics and Systems Analysis*. 2022. **58**(1): 157—163. https://doi.org/10.1007/s10559-022-00445-2

Oleksandr O. Letychevskyi

V.M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine, Kyiv, Ukraine

ORCID: https://orcid.org/0000-0003-0856-9771

MODERN SCIENTIFIC PROBLEMS OF CYBER SECURITY

The article contains an overview of modern problems in cyber security and analyzes the role of scientific research in solving them. In particular, two types of research are distinguished - with the use of an algebraic approach and with the use of neural networks, which belongs to the methods of Artificial Intelligence. Algebraic methods are based on usage of automatic theorem proving and solver programs. These studies are conducted to solve two main problems of cyber security. The first problem concerns the detection of vulnerabilities in software and hardware systems and the assessment of their resistance to intrusions. The second problem is the detection of malicious intrusions in real time. The results of research that help create reliable protection against cyberattacks, which is important in modern circumstances for the protection of systems of critical infrastructure objects, are highlighted.

Keywords: cyber security, cyber attack, algebraic modeling, code vulnerabilities, insertion modeling, formal verification, fuzzy testing, neural networks, machine learning, algebra of behaviors.

Cite this article: Letychevskyi O.O. Modern scientific problems of cyber security. *Visn. Nac. Akad. Nauk Ukr*. 2023. (2): 12—20. https://doi.org/10.15407/visn2023.02.012