УДК004.05

*О.О. Летичевський, В.О. Горбатюк*

# АЛГЕБРАЇЧНИЙ ПІДХІД ТА МЕТОДИ ШТУЧНОГО ІНТЕЛЕКТУ В СУЧАСНИХ СИСТЕМАХ ВИЯВЛЕННЯ ВТОРГНЕНЬ

**Летичевський Олександр Олександрович**

Інститут кібернетики імені В.М. Глушкова НАН України,
orcid: https://orcid.org/0000-0003-0856-9771

*oleksandr.letychevskyi@litsoft.com.ua*

**Горбатюк Віктор Олександрович**

аспірант Інституту кібернетики імені В.М. Глушкова НАН України,
orcid: https://orcid.org/0000-0001-7544-0260

*viktor.gorbatiuk@gmail.com*

З різким зростанням темпу розвитку технологій та впровадженням їх в інфраструктуру підприємств різних сфер діяльності виникає потреба в аналогічному розвитку засобів мережевої безпеки. Сучасні мережі мають багато точок доступу та обробляють великі обсяги трафіку. У зв'язку з залежністю від підключення до інтернет та використання хмарних сервісів, зростає інтерес до приватної інформації в мережі, з'являється все більше підходів до атак, кіберзлочинність стає більш винахідливою у використанні нових технологій. Таким чином, система захисту повинна мати можливість виявляти нові типи вторгнень з високою точністю та мінімальним числом хибних спрацювань. Ключову роль у засобах безпеки грає виявлення вторгнень та можливість заблокувати вторгнення порушника в реальному часі. Активний підхід до кібербезпеки, що дозволяє виявляти потенційні загрози та швидко реагувати, надають системи виявлення вторгнень, що можуть перевіряти мережевий трафік, виявляти шкідливі програми та запобігати будь-якій підозрілій активності. Дана робота має на меті — показати основні види систем виявлення вторгнень та описати їх принцип роботи. Зокрема, з метою знайти систему, що відповідатиме новим поставленим вимогам, у роботі розглянуто два ефективних підходи: метод алгебраїчного співставлення на основі алгебраїчного моделювання та метод виявлення вторгнень з використанням нейронних мереж. Для перевірки та порівняння ефективності обох методів створено прототипи систем, націлених на виявлення однієї з найпоширеніших атак типу «людина–посередник» — ARP Spoofing. За результатами аналізу їх переваг та недоліків запропоновано новий підхід, а саме, комбінацію обох підходів, що включає їх сильні сторони та має перспективи для розвитку.

**Ключові слова:** кібербезпека, кібератака, система виявлення вторгнень, алгебраїчне моделювання, нейронна мережа, алгебра поведінок, агенти та середовища.

## Вступ. Сучасні задачі кібербезпеки та алгебраїчна школа академіка Віктора Глушкова

Алгебраїчна школа академіка В.М. Глушкова дала поштовх до вирішення класичних задач штучного інтелекту та впровадження їх в сучасну індустрію.

Задача формалізації та моделювання дедуктивного мислення, як виведення фактів із певних знань, зумовила появу напряму автоматизації доведення теорем. Дана задача започаткована в Інституті кібернетики ще в 70-х роках минулого сторіччя, в результаті було створено одну з перших у світі машин доведення [1].

Подальший розвиток алгебраїчного напряму привів до створення системи алгебраїчного програмування [2], що працювала на основі мови АПЛАН. Досить велика кількість теоретичних задач в різних сферах математичної науки була формалізована та представлена у вигляді алгебраїчних програм. Дана система взята за основу для вирішення задач верифікації та тестування в довгостроковому проєкті компанії Моторола. Подальший розвиток системи полягав у розширенні можливостей алгебраїчного підходу для паралельних систем. Так було створено систему інсерційного моделювання [3], в основу якої покладено теорію агентів та середовищ, метод алгебраїчного моделювання та алгебра поведінок.

Нині в рамках системи вирішуються задачі, пов'язані з основними викликами кібербезпеки, а саме, виявленням вразливостей в програмних та апаратних системах [4] та виявлення кібератак в мережевому середовищі.

В умовах надзвичайного інтересу до технології нейронних мереж синергія алгебраїчного підходу та сучасних методів штучного інтелекту виявилась досить корисною для вирішення задач кібербезпеки. У даній статті розглянуто методи виявлення атак з використанням обох технологій із перспективою створення систем виявлення вторгнень нового типу.

## 1. Системи виявлення вторгнень

Система виявлення вторгнень (СВВ) — це програмне забезпечення, що є потужним інструментом безпеки для запобігання небажаного доступу до корпоративних мереж, яке слідкує за мережевими пристроями та даними протоколів з метою виявлення відомих шкідливих операцій та неадекватної поведінки щодо порушення політики безпеки, заздалегідь аналізує їх, блокує і видає попередження при виявленні підозрілої активності.

СВВ допомогає прискорити виявлення мережевих загроз, сповіщаючи мережевих адміністраторів про потенційні чи раніше відомі загрози, або автоматизувати весь процес виявлення, надсилаючи дані до централізованого інструменту безпеки, де їх можна поєднувати з даними інших джерел. Система виявлення не може самостійно зупинити загрози безпеці, тому можливості СВВ зазвичай інтегрують або комбінують з системами запобігання вторгненням, що в результаті сприяє виявленню загрози безпеці та автоматичному вживанню заходів їх запобігання.

Однією з найважливіших особливостей у сучасних системах виявлення вторгнень є можливість збору інформації як з вузлів мережі, так і з інших ресурсів, що сприяє розвитку точності виявлення вторгнень і таким чином, з точки зору продуктивності, виявленню більшої кількості загроз з меншою кількістю помилкових спрацьовувань[5].

**Принцип роботи СВВ.** Системи виявлення — це програмні додатки, що можуть бути встановлені на комп'ютерах користувачів або спеціальних апаратних мережевих пристроях. Незалежно від типу системи, всі вони використовують один з двох основних методів виявлення загроз: на основі сигнатур або на основі аномалій [6].

На рис. 1 представлено схему мережі, яка має СВВ, що складається з маршрутизатора, підключеного до інтернет, мережевого екрана та певної кількості кінцевих вузлів (комп'ютерів).

Метод виявлення на основі сигнатур полягає в аналізі мережевих пакетів на наявність унікальних характеристик або поведінки, пов'язаних із конкретною загрозою (сигнатур атак). СВВ цього типу включає базу даних сигнатур атак, з якими вона порівнює мережеві пакети. Якщо пакет збігається з одним із записів бази даних, то система виявлення фіксує це. Щоб зберегти ефективність системи з появою но-

вих кібератак і розвитку відомих атак, бази даних сигнатур необхідно регулярно оновлювати новими даними про загрози. Абсолютно нові атаки, які ще не були проаналізовані, можуть бути не виявленими системою виявлення на основі сигнатур.


Рис. 1

Методи виявлення на основі аномалій використовують машинне навчання. Система виявлення за допомогою моделі нейронної мережі класифікує мережеву активність як нормальну чи підозрілу за рядом ознак, наприклад пропускної здатності чи номеру порту мережевого вузла. Такі системи мають можливість постійного вдосконалення базової моделі нормальної мережевої активності. Оскільки така система спрацьовує при будь-якій аномальній поведінці, то вона часто може відловлювати абсолютно нові кібератаки, пропущені системою виявлення на основі сигнатур, але, як правило, бути більш схильною до помилкових спрацьовувань.

Незалежно від методу виявлення, роботу системи виявлення можна розділити на чотири етапи: попередня обробка, аналіз, відповідь і виправлення. Набір даних мережевого трафіку спочатку попередньо обробляється, а потім оброблені дані аналізуються відповідно до обраного методу, щоб визначити, сталося вторгнення чи це звичайна подія. Під обробкою даних розуміється відбір потрібних для аналізу даних та приведення їх до чітко визначеного формату, включаючи перетворення типів даних та нормалізацію числових значень у певному діапазоні. Наступним етапом є реакція на аналіз даних, що попереджає або сповіщає адміністраторів про аномалії та визначає, яку дію слід виконати у відповідь на викликану подію, і нарешті, — останній етап виправлення, що точно підлаштовує інструмент СВВ відповідно до результатів виявлення, підвищуючи свою ефективність.

## 2. СВВ на основі штучного інтелекту

Ці системи тільки почали з’являтися і, загалом, продовжують розвиватися. Штучний інтелект відіграє важливу роль у виявленні вторгнень і вважається кращим способом адаптації та створення СВВ. У сучасних алгоритмах в задачах кібербезпеки з’являються нові підходи штучного інтелекту, які можна застосовувати для вирішення задач реального часу, а саме, — технологія нейронних мереж.

Ідея використання нейронних мереж у СВВ набула популярності, оскільки конструкція нейронних систем — це гнучкий і ефективний інструмент і може використовуватись у широкому діапазоні екосистем. Наприклад, це може ефективно використовуватись у надзвичайно масштабованих кіберфізичних системах, де

проводиться безперервний аналіз даних у складній глобальній мережі, де є багато пристроїв інтернету речей, які можуть періодично підключатися та відключатися від екосистеми [7].

СВВ на основі нейронних мереж намагається протестувати та вивчити нормальні або типові форми мережевого трафіку, створюваного мережевими пристроями, і виявити аномалії на основі відхилень від цих нормальних або типових форм трафіку. Такі системи можуть бути дуже ефективними, якщо їх правильно спроектувати, але для цього потрібні хороші набори даних для навчання моделі та досвід коригування та налаштування моделі.

Проведено чимало дослідницьких робіт використання мережевих СВВ і штучного інтелекту, але на сьогоднішній день ще немає надійного набору даних, який охоплював би всі можливі мережеві атаки, як добре відомі, так і сучасні. Вибір правильного набору даних та його ознак є важливою складовою виявлення вторгнень. Ознаки — це інформація високого рівня, що задає структуру набору даних. Кожна ознака представляє частину даних одного типу і часто визначається як «змінна» чи «атрибут» даних. Деякі ознаки необхідні для всіх типів атак, тоді як інші потрібні частково або лише для конкретних атак.

Проведено експеримент із виявлення мережевих вторгнень на основі технологій нейронної мережі, для виявлення атаки ARP Spoofing протоколу HTTP, яка є типом атаки «людина посередині». У цій роботі для навчання моделі СВВ використаний набір даних Kitsune [8]. Його зібрано в 2018 році, з оновленням у 2020-му, він містить декілька мільйонів рядків з 115 ознак для дев'яти різних типів атак, включаючи потрібну нам атаку.

Для моделі нейронної мережі використовуватимемо алгоритм багатошарового персептрона (Multi Layer Perceptron — MLP). Персептрон — це лінійний або бінарний класифікатор, одна з базових моделей нейромереж. Персептрон може мати один шар, але зазвичай нейронна мережа — це багатошаровий персептрон. Багатошаровий персептрон чутливий до масштабування ознак, тому для побудови нейронної мережі необхідно масштабувати дані. Масштабування даних — це процес трансформації значень ознак набору даних в межах певного числового діапазону (наприклад, від 0 до 1) з метою покращити продуктивність алгоритму. Багатошаровий персептрон складається з одного вхідного шару з вузлами, що відповідають кількості ознак набору даних, декількох прихованих шарів та одного вихідного. Кожен шар, в свою чергу, має певну кількість вузлів, з'єднаних з виходами попереднього шару, вагові значення для кожного вузлового зв'язку та функцію активації. Основна роль функції активації полягає в перетворенні суми зважених вхідних даних у вихідне значення, щоб надати нейронній мережі нелійних властивостей. Шар персептрона отримує вхідні дані, застосовує певні вагові значення, а вихідне значення створюється активаційною функцією з отриманих зважених вхідних даних. Нейронна мережа моделюється шляхом додавання послідовно кількох шарів персептронів для формування багаторівневих персептронів штучної нейронної мережі. Вхідний рівень безпосередньо приймає дані, а вихідний створює кінцеві результати. Кількість шарів нашої моделі складала 5, включаючи вхідний та вихідний шар та три прихованих шари.

Модель багатошарового персептрона багаторазово навчалась даними з різними кількостями атак. У спробах з усіма типами атак та кількома атаками одного вектора проводилась не тільки бінарна класифікація (розділення трафіку на нормальний та підозрілий), а також і багатокласова, тобто не тільки виявлялась атака, й визначався її тип. Тестування запропонованої моделі для СВВ показало досить непогані результати, точність виявлення або класифікації в більшості з підходів складає 70–85 %.

### 3. Алгебраїчний підхід в СВВ

Алгебраїчний підхід — один з найбільш надійних СВВ. Основним методом запобігання атакам є виявлення вторгнень в систему за допомогою алгебраїчного моделювання. Щоб виявляти атаки в реальному часі, використовуються методи виявлення, в яких відстежуються всі процеси в мережевому середовищі.

**Загальна схема виявлення атаки за допомогою алгебраїчного підходу.** Використовуючи модель мережі в процесі моніторингу її трафіку, в СВВ можна застосовувати метод алгебраїчного співставлення. Для цього для кожного типу атаки необхідно мати відповідний шаблон. Роботу системи виявлення представлено на рис. 2.

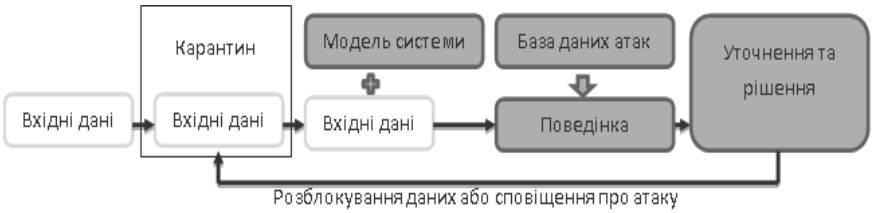


Рис. 2

Процес виявлення атак у реальному часі такої системи полягає у співставленні моделі поведінки мережі в момент отримання вхідного пакета даних з шаблоном атаки. Тому для ефективності роботи системи виявлення важливо мати повну модель системи, що підлягає захисту, та базу даних шаблонів усіх можливих атак.

Система виявлення працює як мережевий екран, тобто затримує весь вхідний трафік на період його перевірки та пропускає лише пакети трафіку, які не були класифіковані як підозріли. В іншому випадку атаки виявлялися б з запізненням вже після успішного проведення.

Система виявлення та запобігання вторгнень — це окремий вузол, що отримує інформацію з зовнішнього середовища та тримає її в карантині. На основі вхідних даних, які потенційно можуть бути шкідливими, за допомогою моделювання генерується відповідна поведінка системи. Далі вона просто порівнюється з усіма доступними в базі даних шаблонами атак.

У випадках, коли знаходиться співпадіння і змодельована поведінка призводить до атаки, надсилається сповіщення про атаку, а відповідний пакет даних блокується остаточно. Якщо ж поведінка не приводить до атаки, то аналізується наступний пакет у такому ж порядку.

Якщо відповідності з шаблонами не знайдені, то такі дані відразу пропускаються. Інколи, щоб не затримувати мережевий потік, дані, що викликали співпадіння, але не привели до атаки, пропускаються.

**Алгебра поведінок, алгебраїчне співставлення.** Алгебра поведінки — досить розвинений математичним інструмент для формалізації як математичних, так і природних явищ у вигляді рівнянь поведінки, тому багато формальних методів розроблені на її основі і використовуються в контексті теорії агентів та середовищ, розробленої О.А. Летичевським та Д. Гільбертом [9].

Агент — це сутність в межах якогось середовища, що своїми діями змінює стан цього середовища. Ідея схожа до понять кінцевих автоматів чи транзиційних систем, де набір поведінкових сценаріїв складає поведінку агента. Стан та тип агента визначається множиною атрибутів певного типу.

Взаємодія агентів в середовищі відбувається завдяки обміну повідомленнями або атрибутів спільного для них середовища. Також можлива багаторівнева дере-

воподібна структура, де агент одного середовища є середовищем для агентів нижчого рівня і відповідно також може бути агентом в середовищі вищого рівня.

Теорія агентів та середовищ визначає інсерційну функцію, що відповідає за появу агента в середовищі, його взаємодію з іншими агентами та виключення його з середовища. Інсерційна функція пердставлена у вигляді рівняння алгебри поведінки.

Алгебра поведінки — це універсальна двосортна алгебра, де основний сорт — це набір поведінок, а другий — набір дій. Алгебра поведінки має операції префіксінгу («.»), недетермінованого вибору («+»), паралельної («||») та послідовної («;») композиції поведінок.

Кожна дія агента складається з:

- передумови дії, представленої формулою;
- процесної компоненти, що ілюструє перехід агента в інший стан;
- післяумови.

Запис виразу поведінки — це послідовність дій та змінних поведінки з операціями алгебри поведінок. Система поведінкових рівнянь представляє собою множину рівностей, в лівій частині яких стоїть змінна поведінки, а в правій — її вираз.

Вирішення системи поведінкових рівнянь передбачає зведення їх до канонічної нормальної форми з розгортанням підстановок та зіставлень виразів поведінки із шаблоном, що містить невідомі поведінки. Розв'язком є поведінка, що відповідає заданому шаблону [10].

Метод алгебраїчного співставлення — це метод пошуку поведінки, що відповідає заданому шаблону в системі рівнянь. Шаблон також може бути представлений у вигляді системи поведінкових рівнянь з набором дій та їхніми передумовами і післяумовами.

Алгебраїчне співставлення проводиться в два кроки:

- знаходження послідовності дій, що відповідає шаблону;
- підтвердження досяжності поведінки з початкового стану.

Перший крок проводиться шляхом рішення поведінкових рівнянь відносно поведінки, представленої в шаблоні, а підтвердження досяжності поведінки — шляхом алгебраїчного моделювання, починаючи від початкового стану та закінчуючи останньою дією шаблону поведінки.

**Шаблони атак.** Вони створюються як послідовності дій в критичних точках, починаючи з вхідних даних зовнішнього середовища і закінчуючи успішною атакою. Послідовність дій, з яких складається поведінкове рівняння шаблону, може бути як лінійне, так і розгалужене. Особливості кожного вторгнення представлені діями з відповідними передумовою та післяумовою. Тому, розуміючи природу атак та послідовність їх проведення, можна створити шаблони для кожного з нині відомих вторгнень.

**Приклад формалізації системи.** Розглянемо приклад ARP Spoofing в рамках формалізації протоколу HTTP та взаємодію агентів в мережевому середовищі. Існує декілька агентів (вузлів), кожен з яких має IP-ім'я та визначається перелічувальним типом IP_NAME, що містить всі можливі IP-імена. Значення атрибута є символьним рядком, наприклад, 192.168.1.1. Відповідно кожен агент має мережеву MAC-адресу, що також визначається множиною перелічувального типу MAC_NAME. Значення атрибута також є символьним, наприклад, 00:00:5e:00:53:af.

Представимо тип агента NODE, який визначається своїми атрибутами:

- IP:IP_NAME — IP-адреса агента.
- list_IP: (int) –> IP_NAME — функціональний атрибут агента, що містить IP-агенти з таблиці, в якій записані адреси всіх агентів, яким колись було надіслано повідомлення, або отримано.

• $M$ — кількість рядків у таблиці або кількість адрес, з якими контактував агент.

• MAC:MAC_NAME — MAC-адреса агента.

• list_MAC: (int) –> MAC_NAME — MAC-адреси всіх агентів у таблиці.

Кожен агент типу NODE має ім'я — $a1$, $a2$, …

При взаємодії агенти виконують дії, що відповідають протоколу обміну повідомленнями. Такими є наступні дії, параметрами яких є значення імен агентів та їх IP-адреси.

SendRequest($x$, $z$) = (Exist $i$:int)($z$ == $x$.list_IP($i$) && (1 <= $i$ <= $x.M$))-> «send Request(x.IP, list_MAC(i))» 1

Дана дія відправляє повідомлення Request, де $x$ — агент відправник, $z$ — адреса отримувача.

У передумові міститься істинність твердження, що адреса знаходиться у списку адрес агента $x$, процесна компонента ілюструє дану дію надсиланням запиту за відповідною до $z$ мережевої MAC-адреси від агента з відповідною $x$ IP-адресою. Післяумови, що визначає зміну атрибутів середовища, немає, тому пишеться 1.

Таким чином, записуються інші дії протоколу HTTP.

GetRequest = (Exist $x$:NODE, $y$:IP_NAME, $u$:MAC_NAME,$i$:int) ($y$ == $x$.list_IP($i$)) && (1 <= $i$ <= $x$.M) && ($u$ == $x$.MAC) -> «receive Request ($y$, $u$), send Response ($x$.IP, $x$.list_MAC($i$))» 1

• агент отримує запит Request ($y$, $u$), де $y$ — IP агента відправника, $u$ — MAC адреса отримувача.

GetResponse = (Exist $x$:NODE, $u$:MAC_NAME, $y$:IP_NAME) ($u$ == $x$.MAC) –> «receive Response ($y$, $u$)» 1

• агент отримує відповідь на запит, а саме, сповіщення Response.

NoSendRequest($x$, $z$) = (Forall $i$:int)($z$ != $x$.list_IP($i$) && (1 <= $i$ <= $x.M$))–> «» 1

• дія агента, в якій $z$ не знаходиться в списку адрес агента $x$ і сповіщення не відправляється.

SendARPRequest($x$, $z$) = (Forall $y$:NODE) –> «send Broadcast($x$.IP, $x$.MAC, $z$)» 1

• дія агента, коли адреса не міститься у списку агента, він надсилає запит до всіх агентів у мережі, щоб ідентифікувати потрібного, та надсилає свою адресу.

GetARPRequest = (Exist $x$:NODE, $y$:IP_NAME, $u$:MAC_NAME, $z$:IP_NAME) ($z$ == $x$.IP) –> «receive Broadcast ($y$, $u$, $z$), send ARPResponse ($x$.IP, $x$.MAC, $u$)» 1

• дія, в якій відбувається отримання Broadcast повідомлення агентом $x$, який отримав запит на свою адресу.

GetARPResponseExist = Exist ($x$: NODE, $y$: IP_NAME, $z$:MAC_NAME, $u$: MAC_NAME, $i$:int) ($x$.MAC == $u$) && ($x$.list_IP($i$) = $y$) && (1<=$i$<=x.$M$) –> «receive ARPRequest ($y$, $z$, $u$)» (list_MAC($i$) = $z$)

GetARPResponseNew = Exist ($x$: NODE, $y$: IP_NAME, $z$: MAC_NAME, $u$: MAC_NAME) (Forall($i$:int) ($x$.list_IP($i$) ! = $y$) && (1<= $i$ <= $x.M$)) && ($x$.MAC == $u$) –> «receive ARPRequest ($y$, $z$, $u$)» ($x$.M = $x$.M + 1; $x$.list_IP($M$ + 1) = y; list_MAC ($M$ + 1) = $z$)

• дія агента, який шукав адресу, де він отримує ARPResponse, та вносить її в свій список.

Поведінкове рівняння, що представляє даний протокол, буде наступною паралельною композицією агентів:

$B0 = B1\ (a1,a2) \parallel B1\ (a1,a3) \parallel \ldots \parallel B1(a2,a1) \parallel (a2,a3) \parallel \ldots$ ,

$B1(x, z)$ = AgentRequest($x$, $z$).$B1$,

AgentRequest1($x$, $z$) = (SendRequest ($x$, $z$.IP).GetRequest.SendResponse. GetResponse + NoSendRequest($x$, $z$.IP).SendARPRequest($x$, $z$.MAC). GetARPRequest. SendARPResponse.(GetARPResponseNew + GetARPResponseExist)

**Шаблон атаки ARP-Spoofing та її виявлення.** Агент-зловмисник може скористатися можливістю надсилати неправдиві дані і вдавати нібито його MAC-адреса підходить під IP-ім'я, що ми запитуємо. Це робиться для того, щоб перехоплювати сповіщення, які направляються цьому агенту.

Тоді шаблон атаки мати вигляд:

*X* = NoSendResponse (*x*, *z*). SendARPRequest (*x*, *z*). GetARPRequest (*x*, *z*)

У даному шаблоні умова порушення правил протоколу представлена дією шаблону GetARPRequest (*x*, *z*). Замінивши MAC-адресу адресата своєю, зловмисник зможе отримувати повідомлення, що надсилатимуться адресату:

GetARPRequest (*x*, *z*) : (Forall *i*:ipnt)(*z* != *x*.list_IP(*i*)) –> »» 1

Тому при обміні повідомленями між агентами типу NODE повідомлення проходитимуть перевірку та не затримуватимуться в карантині до тих пір, поки дія GetARPRequest не з'явиться в моделі поведінки агента. У той же час значення атрибутів x та z матимуть різні значення, що впливає на уточнення шаблону.

Отже, при будь-якому співпадінні моделі поведінки з шаблоном вторгнення повідомлення з підміною MAC-адреси буде заблоковано, а вторгнення зупинене.

## 4. Проблеми в підходах. Недоліки та переваги алгебраїчного та AI-підходу в задачах кібербезпеки

Підходи штучного інтелекту, зокрема технологія нейронних мереж, зустрічаються дедалі частіше у сучасних системах виявлення та запобігання вторгненням у комп'ютерні системи та мережі. Існуючі моделі нейронних мереж досить швидко виявляють аномальну поведінку системи та класифікації її за типом вторгнення, якому піддається атакована система.

Однак технологія нейронної мережі має кілька проблем: висока вірогідність хибних виявлень, змагальні атаки, відсутність якісних, сучасних тренувальних наборів даних і складність класифікації вторгнень, відсутніх у тренувальному наборі даних.

Змагальна атака — це модифіковані дані, які характеризуються практично непомітною зміною невеликої величини тих даних, що найбільшою мірою впливають на класифікацію, і таким чином модель машинного навчання може помилитися [11].

Проблема хибних впливів у системах виявлення вторгнень полягає в тому, що важко визначити нормальний профіль мережевого трафіку, оскільки він змінюється залежно від різних сеансів і різних програм, а також можливого неправильного користування ресурсами. Неправильна класифікація може бути отримана двома способами: система не виявляє атаку, оскільки вона схожа на відомий звичайний шаблон трафіку або коли система виявляє помилкову атаку, оскільки це новий шаблон трафіку, з яким система раніше не стикалася.

Проблема полягає у високій вірогідності помилок, набагато вищій, ніж в інших підходах. Ціна неправильної класифікації атаки при виявленні вторгнень досить висока. Якщо виникає хибний позитивний результат, то це може призвести до дорогих втрат часу аналітика, а з іншого боку, якщо виникає хибний негативний результат, це може завдати серйозної шкоди системі.

Також існує семантична проблема при використанні машинного навчання для виявлення вторгнень. Навіть якщо модель зможе виявити відхилення від нормального профілю, все одно існує необхідність інтерпретувати це відхилення, визначити, наскільки воно шкідливе, і надати результати з точки зору оператора, щоб мати можливість пов'язати виявлення з правильною дією.

Через відсутність наборів даних існують проблеми з оцінкою продуктивності системи. З одного боку, реальні набори даних недоступні для загального використ-

тання через вміст конфіденційних даних, які не можна оприлюднювати, а з іншого боку, змодельовані набори даних недостатньо точні. Також важко розробити надійну схему оцінки систем виявлення вторгнень, оскільки невідомо, як виглядатимуть та поводитимуться майбутні атаки [12].

Використання алгебраїчних методів і сучасних систем вирішення для точного виявлення атак у реальному часі непоширено, оскільки вони набагато повільніші, ніж класифікація нейронних мереж. Алгебраїчний підхід зарекомендував себе досить точним, але не був ефективним з точки зору швидкості. Дані трафіку деякий час залишалися в карантині для пошуку можливого шаблону, критичного для нормальної роботи мережі. До того ж перевірка всього мережевого трафіку алгебраїчним методом — досить ресурсозатратний процес, що виключає можливість розміщення системи виявлення на мережевих пристроях.

## 5. Гібридний спосіб виявлення атак як комбінація технології нейронних мереж та алгебраїчного підходу

При проведенні експериментів з різними підходами до вирішення проблем виявлення мережевих вторгнень виникла потреба створення гібридного методу виявлення атак. Через недоліки алгебраїчного підходу використовуються властивості нейронної мережі, щоб допомогти алгебраїчному методу визначити тип атаки. Технологічна схема процесу представлена на рис. 3.

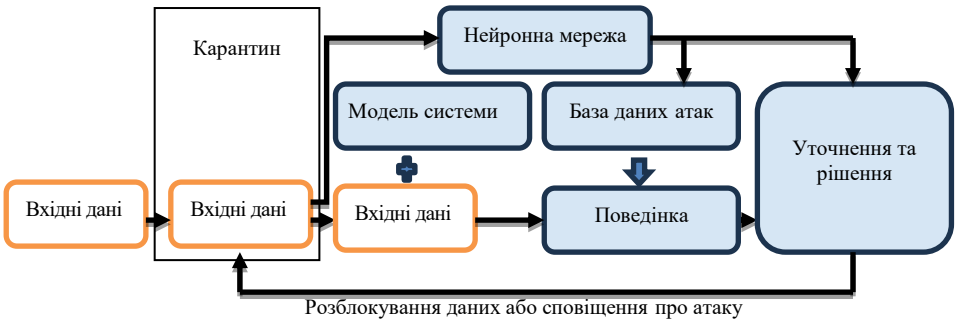


Рис. 3

У даній схемі маємо композицію двох класифікаційних моделей: 1) нейронна мережа, натренована великою кількістю прикладів; 2) алгебраїчний компонент, що містить модель системи, яку захищають, та базу даних шаблонів атак.

Таку композицію моделей назвемо когнітивною мережею (cognitio — пізнання, вивчення, усвідомлення). Цей термін використовується в кількох різних контекстах і пояснюється як здатність сприймати та обробляти зовнішню інформацію та самонавчатись.

Як і в попередньому випадку, вхідні дані на деякий час поміщаються в карантин. Однак у цій схемі нейронна мережа першою сприймає дані і визначає тип атаки. Потім інформація з визначення типу атаки передається до бази даних шаблонів алгебраїчної частину, де активується відповідне завдання алгебраїчного зіставлення, яке підтверджує, що ця атака не була помилковим спрацюванням.

Розглянемо роботу когнітивної мережі в навчальному режимі, коли аналіз і класифікація даних виконуються обома компонентами. Оскільки алгебраїчна модель має високу точність, її можна використовувати для тренування та підвищення надійності нейронної мережі. Це відбувається, коли результати виявлення даними компонентами не співпадають, і такі дані використовуються для

дотренування системи. Зазначимо, що тренувальний набір даних у цьому випадку створюється в реальних умовах роботи системи та на реальних даних. Таким чином, тренування та класифікація відбуваються в однакових умовах.

Такий технологічний підхід реалізований у відповідному програмному науковому прототипі, що пройшов тестування на виявлення атаки ARP-спуфінгу. При комбінованій роботі нейронної мережі та алгебраїчному зіставленні в режимі постійного виклику алгебраїчного підтвердження класифікованої атаки час скоротився у 20 разів. Точність класифікації при послідовній роботі обох компонентів зросла до 90 % і 100 % при паралельній роботі, хоча у другому випадку відповідно збільшився час виявлення.

## Висновок

Використання когнітивної мережі на основі алгоритму алгебраїчного співставлення та простої моделі нейронної мережі показали її високу здатність ефективніше класифікувати атаки в мережевому середовищі та великий потенціал для подальшої розробки.

Розпізнаний авторами прототип працює в двох режимах. Швидший режим — здебільшого працює тільки нейронна мережа, класифікуючи мережеві дані на досить високій швидкості. Алгебраїчний компонент активується тільки у випадках ймовірної атаки.

У режимі навчання когнітивна мережа здатна до самовдосконалення. При розбіжностях між класифікаціями двох компонентів виконується оптимізація нейронної мережі на даних з реального середовища.

До переваг систем даного типу можна віднести модульну структуру, в якій складні обчислення проводяться на боці сервера, що теоретично дозволяє використовувати таку систему на мережевих пристроях типу комутаторів чи маршрутизаторів.

Недоліком технології є досить складний та тривалий процес створення формальних моделей, алгебраїчної складової. Цей процес потребує глибоких знань семантики наявних атак і вразливостей та виконується вручну, що може призводити до помилок.

Подальші дослідження планується зосередити на автоматизації створення шаблонів із використанням методів індуктивного виведення. На даний момент система здатна виявляти тільки один тип атак, тому розширення бази даних шаблонів атак досить актуальне.

Загалом наявний прототип може використовуватися для дослідницької роботи з існуючими СВВ для порівняльного аналізу.

*O. Letychevskyi, V. Horbatyuk*

## ALGEBRAIC APPROACH AND ARTIFICIAL INTELLIGENCE METHODS IN MODERN INTRUSION DETECTION SYSTEMS

**Oleksandr Letychevskyi**

V.M. Glushkov Institute of Cybernetics of NAS of Ukraine, Kyiv,

*oleksandr.letychevskyi@litsoft.com.ua*

**Viktor Horbatyuk**

V.M. Glushkov Institute of Cybernetics of NAS of Ukraine, Kyiv,

*viktor.gorbatiuk@gmail.com*

With the rapid increase in the pace of technology development and its integration into the infrastructure of enterprises in various fields of activity, there is a need for a similar development of network security tools. Modern networks have the large number of access points and handle a lot of traffic. Due to the dependence on Internet connection and the use of cloud services, interest in private information in the network is growing, more and more attack approaches are appearing, and cybercrime is becoming more profitable with the use of new technologies. Thus, the protection system must be able to detect new types of intrusions with high accuracy and a minimum number of false alarms. Intrusion detection and the ability to block attacker's intrusions in real time play a key role in security measures. Intrusion detection systems that can monitor network traffic, detect malware, and prevent any suspicious activity, provide the proactive approach to cyber security that allows you to identify potential threats and to respond quickly. The purpose of this work is to show the main types of intrusion detection systems and to describe their principle of operation. Thus, in order to find a system that meets the new requirements, the paper discusses two of the most effective approaches: the algebraic matching method based on algebraic modeling and the intrusion detection method using neural networks. To approve and compare the effectiveness of both methods, prototypes of detection systems were created to target the one of the most common man-in-the-middle attacks — ARP Spoofing. Based on the results of the analysis of their advantages and disadvantages, a new hybrid approach was proposed — the combination of both approaches which includes all their strengths and has prospects for future development.

**Keywords:** cyber security, cyber attack, intrusion detection system, algebraic modeling, neural network, algebra of behaviors, agents and environments.

## ПОСИЛАННЯ

1. Лялецький О.В. Алгоритм очевидності та системи САД. Минуле та майбутнє. *Кібернетика та системиний аналіз*. 2021. Том 57, № 1. С. 12–20.
2. Kapitonova J. and Letichevsky A. Algebraic programming in the APS system. *Proc. of ISSAC'90, ACM*. USA : New York, 1990. P. 68–75.
3. Letichevsky A., Letychevskyi O., Peschanenko V. Insertion modeling and its applications. *Comput. Sci. J. of Moldova*. 2016. Vol. 24, N 3. P. 357–370.
4. Letychevskyi O. Algebraic methods for detection of vulnerabilities in software systems. *The 9-th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT*'. Ukraine : Kyiv, 2018. P. 24–27.
5. https://www.zenarmor.com/docs/network-security-tutorials/what-is-intrusion-detection-system.
6. https://www.ibm.com/topics/intrusion-detection-system.
7. https://www.iotworldtoday.com/security/tapping-ai-for-intrusion-detection-systems.
8. Kitsune network attack dataset. https://www.kaggle.com/ datasets/ymirsky/network-attack-dataset-kitsune.
9. Gilbert D., Letichevsky A. A model for interaction of agents and environments. *Trends in algebraic development techniques*. D. Bert and C. Choppy. Eds., LNCS, Cham, Switzerland : Springer-Verlag, 1999. P. 311–328.
10. Letychevskyi O., Peschanenko V. Applying algebraic virtual machine to cybersecurity tasks. *IEEE 9th Int. Conf. Sci. Electron., Technol. Inf. Telecommun. (SETIT)*, Hammamet, Tunisia, 2022. P. 161–169. DOI:10.1109/SETIT54465.2022.9875895.
11. https://www.lumenova.ai/blog/understanding-adversarial-attacks-machine-learning/.
12. Shaya Omar. Using machine learning in networks intrusion detection. 2008. P. 1–13. DOI: 10. 13140/RG.2.1.2586.0321.

*Отримано 21.07.2023*