https://doi.org/10.31891/2219-9365-2025-84-15
УДК 004.52:004.75

САВЕНКО Олег
Хмельницький національний університет
https://orcid.org/0000-0002-4104-745X
e-mail: savenko_oleg_st@ukr.net
ДРОЗД Андрій
Хмельницький національний університет
https://orcid.org/0009-0008-1049-1911
e-mail: andriydrozdit@gmail.com
МЕДЗАТИЙ Дмитро
Хмельницький національний університет
https://orcid.org/0009-0004-3247-6406
e-mail: medzatyid@khmnu.edu.ua

КОНЦЕПТУАЛЬНА АРХІТЕКТУРА ОБМАННИХ СИСТЕМ З ПРИМАНКАМИ І ПАСТКАМИ НА ОСНОВІ ПОПУЛЯЦІЙНИХ АЛГОРИТМІВ

Проблема підвищення ефективності виявлення кібератак (КА) та дій зловмисного програмного забезпечення (ЗПЗ) у корпоративних мережах залишається актуальною у зв'язку зі зростанням складності сучасних кіберзагроз. Її багатогранність зумовлена різноманіттям показників, параметрів і характеристик, які можуть підлягати оновленню або удосконаленню та впливати на результативність систем захисту. Перспективним напрямом для підсилення механізмів виявлення атак є модернізація моделей атак і розвиток архітектури обманних систем з приманками і пастками, які здатні формувати адаптивні, гнучкі та самоорганізовані захисні реакції.

Запропонована архітектура обманних систем передбачає здатність до автоматизованого прийняття рішень щодо подальших дій, організації колективної роботи агентів та гнучкого керування приманками й пастками. Такий підхід, у поєднанні з інформацією про доступні мережні ресурси, створює суттєву перевагу над засобами зловмисників. Використання числових характеристик елементів системи дозволяє здійснювати оцінювання їхнього поточного стану та визначати оптимальні подальші кроки під час протидії атакам. Ці характеристики формують основу для ефективного керування поведінкою приманок і пасток у реальному часі.

У межах роботи запропоновано синтезувати популяційні алгоритми, зокрема алгоритм молі і вогню, з архітектурою обманних систем для оптимізації послідовності наступних кроків системи та підтримки її стабільного функціонування за умов тривалих впливів КА та дій ЗПЗ. Показано, що конфігурації мережних вузлів у сценаріях проникнення ззовні або зсередини можуть бути подані у вигляді двох схем представлення, які узагальнюються до єдиної структури або можуть бути інтерпретовані у тривимірному просторі. Такий підхід забезпечує більш гнучке моделювання поведінки системи та підвищує адаптивність її реакцій. Оптимізація за допомогою популяційних алгоритмів дозволяє уникати повного перебору можливих дій, забезпечує швидку збіжність оптимальних рішень у динамічному середовищі та дає змогу перебудовувати послідовність кроків відповідно до змін у корпоративній мережі. Додатково підтримка динамічної активності приманок і пасток гарантує тривалу взаємодію системи з атакуючими впливами та підвищує її стійкість до складних і повторюваних атак.

Проведені експерименти показали перспективність запропонованого напряму досліджень та синтезу в архітектурі обманних систем популяційних алгоритмів для оптимізації вибору ними подальших кроків.

Перспективи подальших досліджень полягають у поглибленні інтеграції популяційних алгоритмів в архітектуру обманних систем з приманками та пастками, а також у розробленні методу організації їхнього функціонування відповідно до запропонованої концептуальної моделі. Результати роботи спрямовані на підвищення рівня кіберзахисту корпоративних мереж через створення інтелектуальних адаптивних систем протидії сучасним КА та діям ЗПЗ.

Ключові слова: корпоративні мережі; комп'ютерні станції; системи обману; популяційні алгоритми; пастка; приманка; зловмисне програмне забезпечення; комп'ютерні атаки; архітектура систем.

SAVENKO Oleg, DROZD Andriy, MEDZATYI Dmytro
Khmelnytskyi National University

CONCEPTUAL ARCHITECTURE OF DECEPTIVE SYSTEMS WITH BAITS AND TRAPS BASED ON POPULATION ALGORITHMS

The problem of increasing the effectiveness of detection of cyber attacks (CA) and actions of malicious software (MAL) in corporate networks remains relevant in connection with the growing complexity of modern cyber threats. Its versatility is due to the variety of indicators, parameters and characteristics that can be updated or improved and affect the effectiveness of protection systems. A promising direction for strengthening attack detection mechanisms is the modernization of attack models and the development of the architecture of deception systems with decoys and traps that are capable of forming adaptive, flexible and self-organized defensive reactions.

The proposed architecture of deception systems provides the ability to make automated decisions about further actions, organize the collective work of agents, and flexibly manage decoys and traps. This approach, combined with information about available network resources, creates a significant advantage over the means of attackers. The use of numerical characteristics of system elements makes it possible to evaluate their current state and determine the optimal next steps when countering attacks. These characteristics form the basis for effective control of bait and trap behavior in real time.

Within the framework of the work, it is proposed to synthesize population algorithms, in particular the moth and fire algorithm, with the architecture of deception systems to optimize the sequence of the next steps of the system and support its stable

functioning under the conditions of long-term effects of the CA and actions of the MAL. It is shown that the configurations of network nodes in the scenarios of penetration from the outside or from the inside can be presented in the form of two representation schemes, which are generalized to a single structure or can be interpreted in three-dimensional space. This approach provides more flexible modeling of system behavior and increases the adaptability of its reactions.

Optimization with the help of population algorithms allows you to avoid a complete enumeration of possible actions, ensures fast convergence of optimal solutions in a dynamic environment, and allows you to rebuild the sequence of steps in accordance with changes in the corporate network. In addition, support for the dynamic activity of baits and traps guarantees long-term interaction of the system with attacking influences and increases its resistance to complex and repeated attacks.

The conducted experiments showed the prospects of the proposed direction of research and synthesis in the architecture of deceptive systems of population algorithms for optimizing their choice of further steps.

Prospects for further research are in deepening the integration of population algorithms into the architecture of deceptive systems with baits and traps, as well as in developing a method of organizing their functioning in accordance with the proposed conceptual model. The results of the work are aimed at increasing the level of cyber protection of corporate networks through the creation of intelligent adaptive systems for countering modern computer attacks and actions of malicious software.

Keywords: corporate networks; computer; detection systems; population algorithms; trap; lure; malicious software; computer attacks; systems architecture.

Стаття надійшла до редакції / Received 30.09.2025
Прийнята до друку / Accepted 02.11.2025

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Організація систем виявлення комп'ютерних атак (КА) та зловмисного програмного забезпечення (ЗПЗ) в корпоративних мережах потребує удосконалення на рівні архітектури [1], особливо при використанні приманок і пасток для покращення протидії зловмисникам. Корпоративні мережі з великою кількістю вузлів та посиленим захистом за рахунок підтримки великої кількості приманок та пасток потребують швидкого та ефективного управління ними.

Приманки можуть використовуватись не тільки для відволікання зловмисників та неправильного розуміння ними хибних об'єктів, але й для того, щоб при проведенні зловмисниками атак перенаправити або продублювати трафік на приманки, які є невидимими зловмисникам, з подальшим аналізом трафіку та прийняття відповідних дій. Також, при цьому можуть бути активізовані пастки в поєднанні з приманками. Приманки можуть бути різними та мати різну будову, тому система повинна оцінити їх та здійснити вибір найбільш ефективних з них. Для здійснення вибору приманок з пастками можуть бути застосовані різні стратегії. Але в цьому процесі важливим є вибір саме таких варіантів приманок з пастками, які б могли в процесі проведення атак, привести не тільки до однієї певної приманки. Взагалі приманки з пастками можуть бути досліджені зловмисниками і дуже багато з них не мають повноцінних наборів правильних відповідей, які повинні імітувати, і тому, вони можуть бути зрозумілими зловмисникам через їх виявлення, а в частих випадках і використані ними для зловмисних дій. Тому, керування приманками з пастками має бути організовано з врахуванням спроможності досліджувати комп'ютерні атаки сукупністю таких об'єктів за певними стратегіями. Перспективним напрямом для дослідження таких стратегій є стратегії, що базуються на використанні популяційних алгоритмів в архітектурі систем виявлення комп'ютерних атак з приманками і пастками.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

У роботі [2] розглянуто тенденції безпеки в хмарних моделях. З врахуванням тенденцій безпеки, було проаналізовано проблеми безпеки, включаючи порушення даних, конфіденційність даних, контрольованість доступу до даних, автентифікацію, неадекватну перевірку, фішинг, розкриття ключів, аудит, збереження конфіденційності та хмарні IoT-додатки.

У роботі [3] розглянуто модель поведінки зловмисників, при якій кіберзлочинці почали використовувати кібератаки як послугу для автоматизації атак та посилення їхнього впливу. Зловмисники використовують вразливості, що існують в апаратному, програмному та комунікаційному рівнях. Різні типи кібератак включають розподілену відмову в обслуговуванні (DDoS), фішинг, атаки типу "людина посередині", атаки за паролем, віддалені атаки, ескалацію привілеїв та шкідливе програмне забезпечення. У роботі розглянуто найновіші атаки, моделі атак та методи виявлення. Встановлено, що використання трендових технологій, таких як машинне навчання, глибоке навчання, хмарні платформи, великі дані та блокчейн, є перспективним рішенням для поточних та майбутніх кібератак.

У роботі [4] запропоновано метод на основі комп'ютерного зору для виявлення зловмисних програм, що знаходяться в основній пам'яті комп'ютера. Він працює, беручи виконувані файли у віртуальному середовищі для вилучення файлів дампа пам'яті з енергозалежної пам'яті та перетворення їх у певний формат зображення. Для покращення контрастності сусідніх пікселів та зменшення ентропії використовуються адаптивне вирівнювання гістограми з обмеженням контрасту на основі комп'ютерного зору та вейвлет-перетворення. Потім використовується метод опорних векторів, випадковий ліс, дерево рішень та класифікатори машинного навчання для навчання моделі на перетворених зображеннях.

У роботі [5] запропоновано та досліджено сім гібридних моделей машинного навчання для виявлення шахрайської діяльності з використанням набору даних реальних текстів. Розроблені гібридні моделі складалися з двох фаз: спочатку для виявлення шахрайства з кредитними картками використовувалися найсучасніші алгоритми машинного навчання, а потім на основі найкращого одинарного алгоритму з першої фази були побудовані гібридні методи.

У роботі [7] розглянуто методи штучного інтелекту, які використовуються для опису характеристик інформації, оскільки вони допомагають у процесі аналізу даних аналізувати дані та виявляти правила та закономірності. Виявлення аномалій є важливою областю, яка допомагає виявляти приховану поведінку в даних, яка є найбільш вразливою до атак. Це також допомагає виявляти вторгнення в мережу. Такі алгоритми, як гібридний масив K-середніх та послідовна мінімальна оптимізація, можуть бути використані для підвищення точності коефіцієнта виявлення аномалій. У роботі представлено модель виявлення аномалій на основі методу машинного навчання.

У роботі [8] розглянуто одну з найважливіших проблем, з якими стикаються користувачі мережі Інтернету. Такою проблемою є ЗПЗ. Поліморфне ЗПЗ – це новий тип ЗПЗ, яке є більш адаптивним, ніж попередні покоління вірусів. Поліморфне ЗПЗ постійно змінює свої сигнатурні риси, щоб уникнути ідентифікації традиційними моделями виявлення ЗПЗ на основі сигнатур. Для виявлення зловмисних загроз або ЗПЗ використано методи машинного навчання.

У роботі [9] розглянуто програмно-визначені мережі (SDN). Це нова мережна парадигма, яка забезпечує централізоване керування, програмованість та глобальний огляд топології в контролері. Розподілена атака відмови в обслуговуванні – це мережна атака, яка переповнює мережеві з'єднання незаконними даними з використанням високошвидкісної передачі пакетів. Нелегітимний трафік даних може перевантажувати мережні з'єднання, що призводить до втрати законних даних та недоступності мережних служб. Низькошвидкісна розподілена відмова в обслуговуванні (LDDoS) – це нещодавня еволюція DDoS-атаки, яка стала однією з найсерйозніших вразливостей для Інтернету, платформ хмарних обчислень, Інтернету речей (IoT) та великих центрів обробки даних. Крім того, атаки LDDoS складніше виявити, оскільки ця атака надсилає велику кількість незаконних даних, замаскованих під законний трафік. В роботі подано проблеми вразливості на всіх рівнях архітектури SDN, які можуть використовувати LDDoS-атаки.

У роботі [10] застосовано підхід машинного навчання (ML) для виявлення мережевих аномалій та створено різні моделі на основі даних для виявлення DDoS-атак. Існуючі методи використовують дані, штучно синтезовані або зібрані з мереж інформаційних технологій або невеликих лабораторних випробувальних стендів. Щоб усунути це обмеження, було використано дані реального мережного трафіку підприємства.

У роботі [11] розглянуто зростання використання комп'ютерних мереж та відповідне збільшення кількості кібератак. СВВ є ключовою технологією безпеки, яка ретельно контролює мережний трафік та активність хостів для виявлення несанкціонованої або зловмисної поведінки. Було розроблено високоточні моделі для виявлення різноманітних кібератак з використанням найменшої можливої кількості ознак, що досягається шляхом ретельного відбору ознак.

У роботі [12] розглянуто виявлення вторгнень у комп'ютерні мережі, яке має велике значення через його вплив на різні сфери зв'язку та безпеки. Виявлення вторгнень у мережу є складним завданням. Більше того, виявлення вторгнень у мережу залишається складним завданням, оскільки для навчання сучасних моделей машинного навчання для виявлення загроз вторгнення в мережу потрібна велика кількість даних. Запропоновано підхід на основі AdaBoost для виявлення вторгнень у мережу на основі вибраних характеристик. На відміну від більшості попередніх досліджень, які використовували набір даних KDD99, було використано комплексний набір даних UNSW-NB 15 для виявлення аномалій у мережі. Цей набір даних являє собою сукупність мережевих пакетів, якими обмінюються хости. Він містить 49 атрибутів, включаючи дев'ять типів загроз, таких як DoS, експлойти, worm-віруси, шелл-коди, розвідувальні атаки, загальні атаки та аналітичні бекдори. Запропоновано AdaBoost на основі класифікатора дерева рішень для класифікації нормальної активності та можливих загроз.

У роботі [13] розглянуто кілька видів атак та можливі наслідки для розуміння ландшафту безпеки в галузі IIoT. У роботі [14] представлено комплексну структуру, спрямовану на формування майбутнього кібербезпеки. Ця структура реагує на складність сучасних кіберзагроз і надає організаціям рекомендації щодо підвищення їхньої стійкості. Основна увага приділяється інтеграції можливостей зі стійкістю. Поєднуючи ці елементи в практику кібербезпеки, організації можуть покращити свою здатність прогнозувати, пом'якшувати, реагувати та відновлюватися після кіберкатастроф. У роботі [15] розглянуто моделі атак, реальні випадки кібервторгнення, що виникли, та превентивні рішення безпеки. У роботі [16] проводиться порівняльний аналіз серед NIDS на основі машинного навчання для розробки орієнтиру різних запропонованих стратегій. Порівнюються як алгоритми поверхневого навчання, такі як дерева рішень, випадкові ліси, наївний байєсівський алгоритм, логістична регресія, XGBoost та методи опорних векторів, так і алгоритми глибокого навчання, такі як DNN, CNN та LSTM, підхід яких є відносно новим і. Також

тестуються ансамблі. Алгоритми оцінюються на наборах даних KDD-99, NSL-KDD, UNSW-NB15, IoT-23 та UNB-CIC IoT 2023.

У роботі [17] наведено цілеспрямовану оцінку поточних кіберзагроз, з якими стикаються університети на основі всебічного огляду доступної інформації. Складено хронологічний графік відомих кібератак на університети, при цьому інциденти класифікуються відповідно до тріади (конфіденційність, цілісність, доступність) та типу інциденту.

Таким чином, для покращення забезпечення безпеки та захисту корпоративних мереж можуть бути застосовані обманні системи з приманками і пастками. Але для їх застосування та покращення обманних технологій необхідно удосконалити архітектуру самих систем, оскільки зловмисники можуть вивчати такі системи протягом тривалого часу і, тому, ці системи повинні мати змінювану поведінку на дії зловмисників.

Метою роботи є покращення функціонування обманних систем з приманками і пастками в корпоративних мережах за рахунок синтезу в їх архітектурі популяційних алгоритмів для заплутування зловмисників та підтримування тривалих відповідей систем на впливи, що викликані КА та діями ЗПЗ.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Архітектура обманних систем для виявлення ЗПЗ та КА в корпоративних мережах з використанням хибних об’єктів для атак

Хибні об’єкти для атак в корпоративних мережах потребують автоматичного організаційного забезпечення. Його може вирішити система виявлення ЗПЗ та КА, яка повинна функціонувати в корпоративних мережах з метою забезпечення їх безпеки та захисту. Певна кількість хибних об’єктів для атак потребує координації роботи зі сторони центру системи. Також, система повинна мати змогу направляти потенційно зловмисні дії безпосередньо на такі хибні об’єкти. Вони повинні мати змогу до здійснення координації виконання завдань і між собою. Основними проблемами у застосуванні систем з хибними об’єктами для атак є спрямування зловмисників саме на такі об’єкти та однакові реакції системи на тривалі повторювані дії зловмисників під час проведення КА.

Хибні об’єкти для атак в корпоративних мережах можуть бути різними. Зокрема: ізольована система, що імітує корпоративну мережу, без зв’язку із захищеною частиною корпоративних мереж і зі зв’язком з її демілітаризованою зоною; підсистема приманок в корпоративних мережах з різними платформами, сервісами, вразливостями без зв’язку із захищеною частиною корпоративних мереж і зі зв’язком з її демілітаризованою зоною; система приманок і пасток, які розміщені перед основними об’єктами захисту; система приманок і пасток, які розміщені поряд з основними об’єктами захисту та імітують їх; зовнішні приманки і пастки, на які йдуть перенаправлення пакетів, що надсилаються в корпоративні мережі; приманки та пастки безпосередньо в комп’ютерних станціях корпоративних мереж, які імітують порти і сервіси тощо. Також, можуть бути інші хибні об’єкти для атак і особливо їх комбінації різні за призначенням та, відповідно, архітектурою. Крім того, хибні об’єкти для атак можуть бути прихованими для зловмисників, в яких система здійснює аналіз спрямованих на них мережних пакетів тощо, тобто тіньові, або такі, що повинні зацікавити зловмисників і це теж є складним завданням для розробників. Така різноманітність та складність хибних об’єктів для атак, а також удосконалення наявних та розроблення нових, вимагають при застосування їх значної кількості забезпечувати ефективну організацію та їх взаємодію. Тому, потребує розроблення система, що повинна функціонувати в корпоративних мережах та мати змогу підтримувати функціонування хибних об’єктів для атак.

Такі системи, як правило, повинні бути обманними, щоб не тільки підтримувати функціонування множини хибних об’єктів для атак, але й самим бути такими, які заплутують зловмисників. Особливо це актуально, за наявності двоцільової моделі КА. Тоді, крім хибних об’єктів для атак зловмисник може шукати і систему, яка здійснює керування ними. В таких системах повинні бути реалізовані обманні технології для приховування їх особливостей від зловмисників. Таким чином, може бути сформований ланцюг кіберобману, що буде підтримуватись такими системами і включатиме самі системи та керовані ними хибні об’єкти для атак.

Зловмисники постійно розробляють нові моделі КА та типи ЗПЗ. Таким чином, в операційному середовищі корпоративних мереж та ззовні потенційні загрози не тільки зростають, але й модифікуються. Відповідями на такі зловмисні дії повинні бути нові технології обману або модифікації відомих технологій, а також їх поєднання.

Для формування ланцюга кіберобману необхідно визначити мету кіберобману, сформувати базу загроз на основі розроблення моделей загроз і моделей КА для їх реалізації, забезпечити збереження історичних даних про поведінку системи та КА під час атак, сформувати методи протидії з використанням обманних технологій, розробити моделі обманних технологій та архітектуру обманних систем, в якій потрібно реалізувати моніторинг подій в корпоративних мережах та керування хибними об’єктами і системою, координацію дій та контроль подій і змінювану поведінку системи та її компонентів.

Реалізація обманних технологій в системі повинна забезпечити введення в оману зловмисників щодо поверхні атаки та, відповідно, приховати критично важливі ресурси корпоративних мереж від зловмисників.

Досягнення таких цілей дасть змогу збільшити ризик виявлення зловмисників при проведенні ними КА, бо змушуватиме направляти або витрачати ресурси на хибні дії та об'єкти в корпоративних мережах. Наприклад, побудувати ланцюг кіберобману можна здійснюючи реконфігурацію мереж, ресурсів та інструментів захисту. При цьому, необхідно змішувати дійсну та фальшиві поверхні атак.

Таким чином, актуальність полягає у синтезі обманних систем протидії та виявлення ЗПЗ та КА в корпоративних мережах, які повинні бути адаптивними, гнучкими, автономними, здатними до колективної взаємодії та прийняття рішень без потреби залучення адмінстарторів. Одним з напрямів розроблення таких обманних систем є реалізація в їх архітектурі популяційних алгоритмів, особливо в частині взаємодіїі з приманками і пастками, а також функціонуванні самих систем. Застосування в архітектурі обманних систем популяційних алгоритмів дає змогу створити динамічне, адаптивне середовище, яке здатне ефективно виявляти нові, невідомі або модифіковані загрози. Основна ідея полягає в тому, що такі алгоритми імітують принципи природного добору, використовуючи популяцію рішень, яка еволюціонує, поступово знаходячи оптимальні або наближені до оптимальних конфігурації системи. У контексті обманних систем для протидії та виявлення ЗПЗ і КА популяційні алгоритми, як клас алгоритмів, можуть бути імплементовані на декількох рівнях архітектури. Це може забезпечити підтримку **самоадаптивних можливостей системи.** Вони можуть постійно запускатись у фоновому режимі для пошуку нових моделей класифікації або адаптації вже існуючих до змін у мережній поведінці. Такий підхід забезпечує стійкість системи до еволюції атак і дозволяє уникати ситуацій, коли обманні системи втрачають актуальність через статичну модель.

В архітектурі, яка побудована на основі мультиагентних технологій, популяційні алгоритми можуть бути локалізованими в окремих агентах, що дозволяє кожному з них адаптувати власну поведінку, а результати еволюції можуть передаватися іншим агентам у рамках спільного навчання. Це створює умови для децентралізованої, гнучкої та масштабованої обманної системи, яка здатна функціонувати у складних динамічних мережних середовищах.

З точки зору архітектури, особливу цінність становить інтеграція популяційних алгоритмів в середовища, які побудовані із застосуванням мультиагентних технологій. У такій архітектурі кожен агент є окремою автономною одиницею, яка здатна виконувати локальну оптимізацію своєї поведінки залежно від отриманої інформації про стан середовища. Розміщення такого модуля безпосередньо всередині агента забезпечує **локальну адаптивність**, коли кожен агент може змінювати власні правила виявлення або стратегії взаємодії з іншими елементами обманної системи. Це важливо у великих або розподілених мережах, де централізоване оновлення моделей є технічно складним або ресурсоємним. Механізм **кооперативного навчання** між агентами дозволяє ефективно передавати нові сформовані знання від одного агента до інших, сприяючи формуванню узагальненої моделі загроз на основі локальних спостережень. Такий підхід забезпечує **децентралізацію,** зменшує залежність від центрального контролера та підвищує загальну стійкість системи до відмов чи цілеспрямованих атак на окремі вузли.

Представимо обманну систему з приманками і пастками в корпоративних мережах для виявлення КА та ЗПЗ в частині саме приманок і пасток так:

$$S = \bigcup_{i=1}^{n} S_i, \tag{1}$$

де n – це кількість приманок і пасток в системі; $n \geq 2$, S_i - i-та приманка чи пастка.

Кількість приманок і пасток в обманній системі з приманками і пастками повинно бути не менше двох, оскільки при невідповідності цій умові система вироджується в одну приманку в комп'ютерній мережі і не зможе виконувати поставлені завдання у решті вузлів в мережі, крім комп'ютерної станції в якій вона розміщена, а також для керування приманками і пастками завдання зводиться до керування одним компонентом обманної системи. Позначення i-тої приманки S_i означає наявність в її складі комп'ютерної станції та програмного забезпечення приманки та/або пастки і системи керування. Тобто кожна приманка чи пастка системи міститься в окремому вузлі в мережі. При цьому допускатимемо, що приманка чи пастка може не мати активного спеціалізованого функціоналу, або активованих засобів, тобто бути нуль-приманкою за спеціалізованим функціоналом. Але вона обов'язково міститиме функційну частину, яка поєднуватиме її з системою або матиме додатково функціонал з керування системою. Особливості функційних можливостей приманки залежать від відповідного наповнення і можуть відрізнятись від решти приманок в системі.

Окремо сформуємо та задамо безпосередньо обманну систему без компонентів, які означені за формулою (1) так:

$$S_{o,osn} = \bigcup_{i=1}^{k} S_{o,osn_i}, \tag{2}$$

де k – це кількість компонентів та модульних елементів основної частини обманної системи; $k \geq 1$, S_{o,osn_i} - i-тий компонент або модульний елемент; $i = 1,2, \dots, k$.

Тоді, обманні системи з приманками і пастками задамо згідно формул (1) і (2) так:

$$S_o = S_{o,osn} \cup S = \cup_{j=1}^{k} S_{o,osn_j} \cup \cup_{i=1}^{n} S_i, \quad (3)$$

де $S_{o,osn}$ – основна частина обманних систем без приманок і пасток; S – частина обманних систем саме з приманками і пастками; n – це кількість приманок і пасток в системі; $n \geq 2$, S_i - i-та приманка чи пастка; k – це кількість компонентів та модульних елементів основної частини обманної системи; $k \geq 1$, S_{o,osn_j} - j-тий компонент або модульний елемент; $j = 1,2,\dots,k$.

Загальна архітектура обманних систем з приманками і пастками згідно з формулами (2) та (3) зображена на рис. 1.

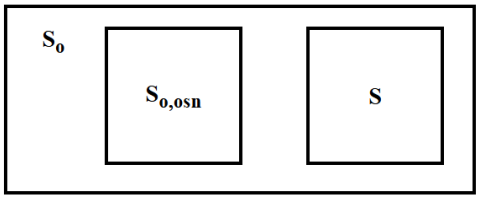

Рис. 1. Загальна архітектура обманних систем з приманками і пастками

Обманні системи S_o повинні мати не менше одного компоненту в основній частині, тобто $k \geq 1$ в формулі (3), і не менше двох приманок та/або пасток, тобто $n \geq 2$ в формулі (3). Таким чином, розглядатимемо обманні системи з мінімум одним компонентом основної частини та мінімум двома приманками чи пастками.

Місце обманної системи в загальній архітектурі корпоративних мереж зображено на рис. 2. В ній виділено чотири типи вузлів в залежності від їх функційного призначення та ступеня залучення компонентів і елементів обманних систем:

1) комп'ютерні станції $KC_{1,0} - KC_{n_{0,0}}$ (n_0 – кількість комп'ютерних станцій), в які встановлені компоненти обманної системи S_o тільки для здійснення керування ними в критичних ситуаціях;

2) комп'ютерні станції $KC_{1,1} - KC_{n_{1,1}}$ (n_1 – кількість комп'ютерних станцій), в які встановлені компоненти підсистеми S обманної системи S_o і вони виконують функції паралельно із завданнями користувачів;

3) комп'ютерні станції $KC_{1,2} - KC_{n_{2,2}}$ (n_2 – кількість комп'ютерних станцій з приманок і пасток) з приманками і пастками, в які встановлені компоненти підсистеми S обманної системи S_o і вони виконують винятково функції приманок і пасток, зокрема і імітують роботу серверів в корпоративній мережі;

4) сервери $C_1 - C_{n_3}$ (n_3 – кількість серверів, в які встановлені компоненти обманної системи S_o тільки для здійснення керування ними в критичних ситуаціях.

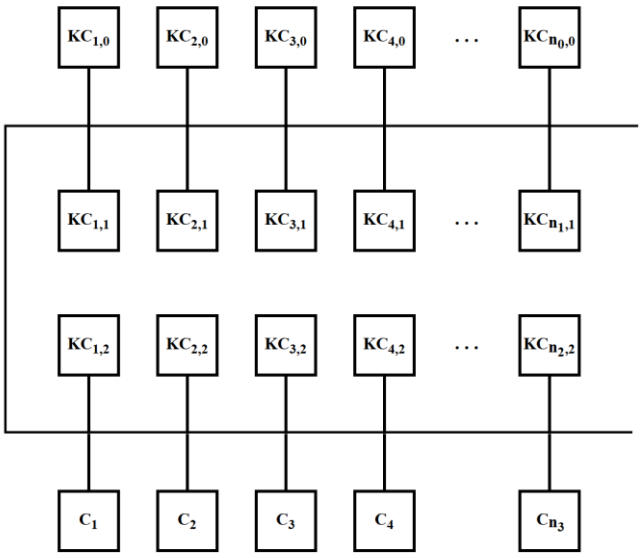

Рис. 2. Загальна архітектура обманних систем з приманками і пастками

Таким чином, зловмисник згідно моделі двоцільової атаки повинен класифікувати вузли комп'ютерних мереж для формування чотирьох класів об'єктів в них.

Центр прийняття рішень, який належить основній частині обманних систем $S_{o,osn}$ (формула (2)), в системах буде знаходитись в одній з компонент системи і може бути переміщений системою в залежності від поточного стану функціонування та вибору подальших кроків в функціонуванні всієї системи. Обманна система з приманками і пастками може мати децентралізовану архітектуру, але тоді зростатимуть витрати часу на обробку подій в ній за рахунок зростання кількості встановлених зв'язків в ній. Тому, розглядатимемо систему з центром, який може переміщуватись системою між різними комп'ютерними станціями в мережі. Для його захисту використовуватимемо як можливість його переміщення так і попередню обробку мережних пакетів в інших вузлах мережі, які надходитимуть до вузла, в якому перебуватиме центр системи.

Синтез обманних систем з приманками і пастками в корпоративних комп'ютерних мережах для виявлення КА та ЗПЗ, в якому буде поєднання системи, мультиагентних технологій та приманок і пасток для виявлення КА і ЗПЗ дозволить отримати переваги над КА і ЗПЗ за рахунок залучення до виявлення та протидії більшої кількості різнотипних обчислювальних засобів та ресурсів порівняно з можливостями зловмисних проявів.

Здійснення керування обманною системою з приманками і пастками в корпоративних комп'ютерних мережах і її компонентами забезпечуватиметься засобами з використанням компонентів штучного інтелекту, бо потребуватиме оперативного вирішення використання приманок і пасток в поточний момент часу, в якому вузлі міститиметься центр системи і куди його за потреби перемістити, як перебудовувати динамічно систему і які подальші дії вона повинна виконувати. Ці завдання включатимуть необхідність перевірки значної кількості параметрів як в системі так і ззовні неї, що потребуватиме оперативної обробки таких параметрів та подій, які впливатимуть на зміну параметрів ззовні.

Необхідність залучення в обманну систему з приманками і пастками в корпоративних комп'ютерних мережах мультиагентних технологій та реалізації створення та керування колективом агентів пов'язано з особливостями КА, частина з яких орієнтована на суттєве перевантаження обчислювальних ресурсів, що потребуватиме розподілу засобів виявлення системи між зловмисними процесами в мережі, а також для виявлення ЗПЗ, яке може перебувати в різних вузлах в мережі та в різних місцях у певному вузлі. Тому, наявність агентів, які кооперуватимуться для вирішення задач з виявлення, покращить ефективність та достовірність результату з виявлення.

Реалізація в обманних системах з приманками і пастками в корпоративних комп'ютерних мережах приманок і пасток різного типу та призначення дасть змогу врахувати використання зловмисниками різноманітних технологій здійснення КА та ЗПЗ. Різні функційні можливості в приманках і пастках можуть бути представлені повністю або частково в певних з них і система визначатиме, які з них будуть залучатись в процес виявлення.

Обманні системи з приманками і пастками для заплутування зловмисника на протязі тривалого часу функціонування повинні мати можливості для перебудови своєї архітектури в динамічному режимі без втручання користувача, тобто в архітектурі систем мають бути закладені механізми забезпечення адаптивності та гнучкості.

Після певного часу функціонування системи з приманками і пастками потребуватимуть визначення своїх подальших кроків. Вони можуть продовжувати виконувати поточні завдання щодо моніторингу процесів в комп'ютерних системах і мережах. Але при виявленні підозрілих подій, або після їх опрацювання, або після динамічної зміни їх архітектури їм потрібно буде визначитись щодо своїх подальших кроків. Для виконання цього завдання вони не повинні очікувати команд від адміністратора чи користувача, бо це сповільнить їх роботу і може призвести, також, до втрати зловмисних проявів, які відбуватимуться протягом часу очікування. Тому такі системи повинні мати можливість здійснювати самоорганізацію, тобто бути самоорганізованими. Побудова механізмів самоорганізації повинна базуватись не на вичерпному переліку станів, в які вони могли б переходити в залежності від опрацьованих подій протягом певного часу, а мати перелік напрямків вибору завдань, які їм потрібно виконати. Щодо детальніших кроків та залучення необхідних засобів для виконання таких завдань, то це вже має вибудовуватись системою самостійно, виходячи з результатів оцінювання наявних засобів для досягнення мети. Обманні системи з приманками і пастками повинні самостійно оцінити можливості виконання завдань з визначеного напрямку свого подальшого спрямування і за потреби, наприклад неможливості виконати рух у визначеному напрямку, який було визначено ними, перебудувати себе та визначити інший напрям для досягнення виконання завдань.

В обманних системах з приманками і пастками повинно бути декілька рівнів, на яких реалізується підтримка прийняття рішень. Це може бути реалізовано в складі одного центру, або може бути в одному центрі і для виконання інших завдань можуть бути використанні компоненти системи, в яких теж знаходитимуться ці підсистеми підтримки прийняття рішень. Ефективність результату виявлення КА і ЗПЗ суттєво залежатиме від архітектури підсистеми підтримки прийняття рішень.

До засобів обманних систем з приманками і пастками для виявлення КА і ЗПЗ необхідним є залучення агентів та організація їх колективної роботи щодо поставлених завдань із виявлення. Наявність таких агентів збільшуватиме можливості системи за рахунок створення цих агентів в динамічному процесі в залежності від поточних потреба та завдань.

На відміну від засобів виявлення КА і ЗПЗ, які орієнтовані на виявлення та негайну реакцію на результат виявлення, засоби побудовані з використанням приманок і пасток орієнтовані саме на продовження подій та дозвіл на проникнення в розставлені системою приманки і пастки для забезпечення захисту важливих ресурсів користувача та отримання максимальної інформації про зловмисні прояви. Це впливатиме на їх внутрішню архітектуру.

В результаті S_i – приманка чи пастка ($i = 1, \dots, n$, n – кількість приманок чи пасток) міститиме такі складові елементи, які задамо вектором:

$$S_i = \left(s_{i,0}, s_{i,1}, s_{i,2}, \dots, s_{i,k}\right), \tag{4}$$

де $s_{i,0}$ – компонента вектора S_i, в якій міститься загальний функціонал компоненти, що відповідає за її функціонування, збір нових даних, комунікацію та функціонування центру системи за потреби; $s_{i,j}$ - j-та компонента вектору S_i, яка показує наявність складової частини компоненти приманки S_i, k – кількість спеціалізованих засобів приманки для ЗПЗ та КА, $j = 1, \dots, k$.

Компоненти вектору S_i ($i = 1, \dots, n$, n – кількість приманок та пасток), які відповідають за складові приманок чи пасток присутні завжди, навіть якщо безпосередні складові компоненти відсутні. В цьому випадку така складова компоненти прирівнюється до нуля: $s_{i,j} = 0$ ($j = 1, \dots, k$).

Таким чином, в обманних системах з приманками та пастками для заплутування зловмисників з метою виявлення КА та ЗПЗ необхідно синтезувати засоби обманних систем, мультиагентних технологій та приманок і пасток. Архітектура таких синтезованих обманних систем повинна надавати можливість будувати засоби, які будуть адаптивними, гнучкими, самоорганізованими, прийматимуть рішення щодо своїх подальших кроків та щодо виявлення, а також організовуватимуть колективну робота агентів. Така архітектура обманних систем з приманками і пастками в поєднанні з відомостями в ній про наявні ресурси комп'ютерних мереж дасть перевагу над засобами зловмисників.

Синтез обманних систем з приманками та пастками виявлення ЗПЗ та КА в корпоративних мережах

Обманні системи з приманками та пастками для забезпечення переваг над засобами зловмисників повинні мати архітектуру, яка надаватиме їм можливості бути адаптивними, гнучкими, самоорганізованими. Для забезпечення таких функційних можливостей застосуємо поділ системи на компоненти і елементи її таким чином, щоб вона досягла певного рівня модульності в залежності від рівнів функціонування та завдань відповідних рівнів. Це важливо для відокремлення функцій з самостійного керування роботою всієї системи і виконання локальних завдань та безпосередньо з реалізації завдань спеціалізованого призначення для організації приманок та пасток і обробки результатів їх роботи. Таким чином, необхідним є введення в модель архітектури системи рівнів та модульності. Зображення архітектури підсистеми S (формули (1), (4)) системи S_o з врахуванням розглянутих вимог задано на рис. 3.

Позначення елементів, які зображено на рис. 3, відповідає змінним в формулах (1) та (4).

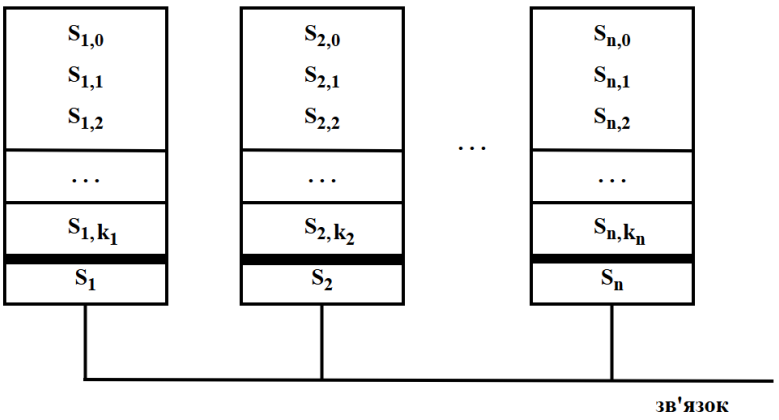

Рис. 3. Узагальнена архітектура підсистеми *S*

Обов'язковими підсистемами елементу S_i ($i = 1, \dots, n$, n – кількість приманок і пасток) множини S є підсистема керування компонентою S_i ($i = 1, \dots, n$), засоби комунікації в системі S, підсистема сенсорів для отримання інформації про зовнішні по відношенню до компоненти події, процеси і стани в комп'ютерній станції, в яку вона встановлена, включаючи також інформацію щодо зовнішніх та внутрішніх комунікацій в середовищі, де експлуатується вся система S, підсистема керування всією системою S за потреби. Таким чином, в узагальненій архітектурі обманних систем з приманками і пастками виділено рівні, на яких розміщено відповідні функційні засоби систем та, відповідно, досягнуто певної модульності в компонентах

системи. Така внутрішня архітектурна організація системи дасть змогу ефективніше змінювати її конфігурацію протягом часу експлуатації.

Розглянемо детальніше організацію функціонування підсистеми керування обманними системами з приманками і пастками з врахуванням сформульованих до них вимог бути адаптивними, гнучкими, самоорганізованими. Підсистема керування обманними системами з приманками і пастками може бути активною в одній з компонент системи, а в решті компонент або неактивною або відсутньою. Визначення наступної компоненти, в якій буде функціонувати підсистема керування системою S, здійснюватиметься в компоненті, в якій знаходиться активна в поточний стан система компонента, або більшість компонент системи визначають наступну таку компоненту при втраті зв'язку з попередньою компонентою, в якій функціонував центр системи і був активним та основним. Крім того, така підсистема повинна здійснювати опрацювання подій, що надходять ззовні системи S від сенсорів, від комп'ютерної станції, в яку вона встановлена як компонент системи S, приймати рішення щодо використання приманок та пасток компоненти і системи в цілому, а також здійснювати планування та визначення своїх подальших кроків без втручання користувача чи адміністратора системи, що є актуальним враховуючи специфіку завдань системи S. Поведінка системи повинна бути незрозумілою як адміністратору системи, так і зловмисникам чи ЗПЗ, інакше результативність її роботи суттєво знизиться. Тобто, вона не повинна бути такою, що на певні початкові набори вхідних даних реагує постійно однаково і передбачувано. Крім того, її подальші дії повинні визначатись нестандартно і бути непрогнозованими. Все це суттєво ускладнює її внутрішню будову підсистеми керування. Таким чином, підсистема керування має враховувати такі вимоги:

1) керувати опрацюванням подій, які надходять ззовні системи;
2) здійснювати комунікацію між розподіленими компонентами системи;
3) підтримувати цілісність системи та змінювати місце розміщення центру системи;
4) розподіляти завдання між підсистемами керування компонентами;
5) здійснювати аналіз результатів роботи приманок і пасток та залучати різні компоненти системи для збільшення та нарощування обчислювальних ресурсів і можливостей з виконання спеціалізованих завдань, які забезпечуються відповідними приманками;
6) визначати подальші кроки у функціонуванні системи;
7) здійснювати аналіз попередніх рішень і згідно його результатів приймати рішення;
8) володіти засобами для визначення нестандартних подальших дій;
9) оцінювати прийняті рішення з керування системою;
10) гнучко перебудовувати свою архітектуру;
11) здійснювати самонавчання протягом всього часу експлуатації системи;
12) приймати оперативні рішення в нестандартних ситуаціях, обираючи оптимальне рішення з можливих варіантів.

Всі ці вимоги відносяться винятково до організації функціонування підсистеми керування системою S і не відносяться до решти підсистем. Наповнення функційних можливостей решти підсистем здійснюватимемо з врахуванням їх завдань та команд отриманих від підсистеми керування. Узагальнена схема з місцем центру керування системою, компонентами центру та відображенням комунікації із зовнішнім середовищем зображено на рис. 4. Узагальнену структуру керування підсистемою S системою S_o з врахуванням функціональних елементів зображено на рис. 5.

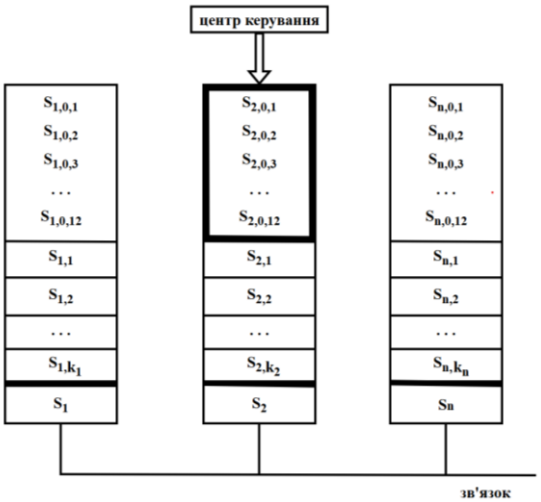

Рис. 4. Узагальнена структура з відображенням місця керування підсистемою *S* та комунікації із зовнішнім середовищем

функційний елемент підсистеми керування для обробки подій, які надходять ззовні системи	$s_{i,0,1}$	компоненти підсистеми: $S_1, S_2, S_3, \ldots, S_n$
функційний елемент підсистеми керування для здійснення комунікації між розподіленими компонентами системи	$s_{i,0,2}$	
функційний елемент підсистеми керування для підтримки цілісності системи та зміни місця розміщення центру системи	$s_{i,0,3}$	
функційний елемент підсистеми керування розподілом завдання між підсистемами керування компонентами	$s_{i,0,4}$	
функційний елемент підсистеми керування для здійснювати аналізу результатів роботи приманок та залучення різних компонентів системи для збільшення та нарощування обчислювальних ресурсів і можливостей з виконання спеціалізованих завдань, які забезпечуються відповідними приманками	$s_{i,0,5}$	
функційний елемент підсистеми керування для визначення подальших кроків у функціонуванні системи	$s_{i,0,6}$	
функційний елемент підсистеми керування для здійснення аналізу попередніх рішень і згідно його результатів прийняття рішення	$s_{i,0,7}$	
функційний елемент підсистеми керування для визначення нестандартних подальших дій	$s_{i,0,8}$	
функційний елемент підсистеми керування оцінювання прийнятих рішень щодо керування системою	$s_{i,0,9}$	
функційний елемент підсистеми керування гнучкою перебудовою архітектури системи	$s_{i,0,10}$	
функційний елемент підсистеми керування для здійснення самонавчання протягом всього часу експлуатації системи	$s_{i,0,11}$	
функційний елемент підсистеми керування для прийняття оперативних рішень в позаштатних ситуаціях	$s_{i,0,12}$	

Рис. 5. Узагальнена структура керування підсистемою S з врахуванням функціональних елементів

Кожна компонента системи містить підсистему керування, тому зображені на рис. 5 функційні елементи керування підсистемою S системи S_o відносяться до всіх компонентів системи $S_1, S_2, S_3, \ldots, S_n$, які формують підсистему S. Функційні елементи $s_{i,0,1} - s_{i,0,12}$ $(i = 1,2, \ldots, n)$ відносяться до підсистеми керування системою і не відносяться підсистеми керування компонентою. Для керування компонентою системи S використовуватимуться інші функційні елементи з цієї підмножини для компоненти $s_{i,0}$ $(i = 1,2, \ldots, n)$. Виділення функційних елементів до такого рівня деталізації дасть змогу досягти модульності, розподілу за рівнями пріоритетності при виконанні завдань і буде використано при визначенні наступного місця розміщення центру керування за рахунок встановлення активності відповідного елементу. Забезпечення інформації щодо активності / неактивності функційних елементів $s_{i,0,1} - s_{i,0,12}$ $(i = 1,2, \ldots, n)$ компонент здійснюватимемо формуванням вектора функційних елементів підсистеми керування компонент системи S з компонентами, які відповідатимуть предикативним функціям. Сформуємо такий вектор, позначивши його $v_{s_{i,0}}$ $(i = 1,2, \ldots, n)$, так:

$$v_{s_{i,0}} = \left(p(s_{i,0,1}), p(s_{i,0,2}), \ldots, p(s_{i,0,12})\right), \tag{5}$$

де $p(s_{i,0,q})$ – предикат, що вказує на активність / неактивність функційного елементу видаючи значення $\{1\}$ або $\{0\}$ відповідно; $i = 1,2, \ldots, n$, n – кількість компонент системи S; $q = 1, 2, \ldots, 12$.

Таким чином, сформована в поточний момент часу така інформація щодо активності / неактивності компонентів буде використовуватись для прийняття рішення системою щодо подальших кроків. Крім того, подальша деталізація керування підсистеми S буде враховувати компоненти сформованого згідно формули (5) вектору. Для кожної компоненти системи сформовані вектори поєднаємо у вектор $V_{s_{0,1}}$ за формулою так:

$$V_{s_{0,1}} = \left(v_{s_{1,0}}, v_{s_{2,0}}, \ldots, v_{s_{n,0}}\right), \tag{6}$$

де n – кількість компонент системи S.

В антивірусних засобах, які використовують приманки і пастки, крім підсистеми визначення зловмисної активності в комп'ютерних системах і мережах, важливою складовою є підсистема керування всією мережею приманок і пасток із залученням різноманітних засобів. Відомостей про підсистему прийняття рішень відомих мереж приманок і пасток недостатньо для їх використання. Крім того, рівень

непередбачуваності дій антивірусних засобів при використанні відомих або готових рішень саме в підсистемі керування усією системою повинен бути належним та непрогнозованим. Тому, цей рівень, на якому здійснюється керування всією системою є важливим і потребує розроблення.

Позначимо зовнішні впливи на компоненти системи і систему S_o в цілому елементами множини W. Позначимо ці впливи узагальнено, без деталізації особливостей та їх класифікації. Наявність впливів в загальній моделі функціонуючих процесів в середовищі розміщення і експлуатації підсистеми S є необхідним, бо такі впливи спонукатимуть систему до реакції на них і відповідно формуватимуть множину наслідків. Врахування впливів, що надходять із зовнішнього середовища, може бути пов'язане як із завданнями для встановлення дій ЗПЗ чи КА, так і дій, які не відносяться до зловмисної активності. Але такі впливи із зовнішнього середовища спонукатимуть підсистему S до їх аналізу та обробки. Крім того, впливи можуть бути викликані і подіями, що відбуватимуться безпосередньо в компонентах самої системи S_o, тобто ці впливи будуть внутрішніми і викликаними в результаті функціонування безпосередньо самої системи S_o. До опрацювання подій, що виникатимуть через внутрішні впливи на компоненти системи S_o, не залучатимуться засоби виявлення ЗПЗ та КА. Але такі впливи можуть відбуватись синхронно із зовнішніми впливами і оскільки система S_o має розподілену архітектуру, то вони повинні враховуватись, бо оброблятимуться в компонентах системи. Також, ці внутрішні впливи можуть бути результатами обробки зовнішніх впливів, оскільки система S_o орієнтована на організацію функціонування мережі приманок для ЗПЗ та КА. Тому, враховуючи складність та природу впливів, яка пов'язана із зовнішніми та внутрішніми джерелами їх появи, розглядатимемо їх всі як елементи однієї множини $W = \{w_1, w_2, \dots, w_{k_S}\}$, де k_S – кількість впливів. До елементів множини W відноситимемо тільки ті впливи, які сприймає система S_o і реагує на них. Можуть бути впливи, на які система S_o не реагує і на які не має засобів сприймати їх. Вважатимемо, що системо S_o матиме засоби для сприйняття максимально можливої кількості впливів. Якщо ж протягом тривалого часу функціонування системи S_o вона спочатку сприймала і обробляла певні конкретні впливи, а в подальшому ні, то такі впливи вважатимемо елементами множини W. Втрата можливості сприйняття частини впливів може бути пов'язана з деградацією системи S_o і підсистеми керування нею повинні обробляти такі події. Також, в середовищі функціонування системи S_o можуть відбуватись процеси, впливи яких не сприйматимуться її засобами, але через певний час її функціонування частина таких впливів буде видима для засобів системи S_o і вони будуть оброблятись нею, тоді їх буде додано до множини W системою S_o.

Елементи множини W можуть бути одночасно ідентифіковані різними компонентами системи S_o. Тому, впливи на систему S_o можуть бути одні і ті ж, але їх спрямування може бути в різних комп'ютерних системах в компонентах системи. В зв'язку з можливістю таких подій, як не одиничних, необхідно враховувати їх при проєктуванні системи S_o. Джерела таких впливів можуть бути різними, а може бути одне. Пошук джерел впливів в комп'ютерних системах в мережі, в які встановлені компоненти системи S_o теж є її завданням, тому аналіз таких впливів і їх джерел є необхідними. Задамо потенційні впливи на компоненти системи S_o з множини W вектором впливів на компоненту так:

$$w_{S_i} = \left(w_{1,S_i}, w_{2,S_i}, \dots, w_{k_S,S_i}\right), \tag{7}$$

де w_{l,S_i} – активність / неактивність l-го впливу на S_i компоненту системи, який кодується значеннями $\{1\}$ або $\{0\}$ відповідно; $i = 1,2, \dots, n$, n – кількість компонент системи S; $l = 1, 2, \dots, k_S$, k_S – кількість можливих впливів.

Тоді, вектор впливів на всю систему S_o в частині приманок і пасток задаватимемо так:

$$W_S = \left(w_{S_1}, w_{S_2}, \dots, w_{S_n}\right), \tag{8}$$

де w_{S_i} – вектор впливів на S_i компоненту системи S; $i = 1,2, \dots, n$; n – кількість компонент підсистеми S.

Згідно формул (7) та (8) таке задання двох типів векторів W_S та w_{S_i} $(i = 1,2, \dots, n)$ дає змогу отримати їх прямий добуток, який формуватиме матрицю впливів на систему S_o в цілому з деталізацією впливів на кожну компоненту. Задамо її формулою так:

$$M_{WS} = W_S \times w_{S_{i(i=1,2,\dots,n)}}, \tag{9}$$

де w_{S_i} – вектор впливів на S_i компоненту системи (формула 1); $i = 1,2, \dots, n$, n – кількість компонент підсистеми S.

Аналогічно за формулами (5) і (6) отримуємо матрицю $M_{V_{S_{0,1}}}$ впливів на функційні елементи компонентів підсистеми S, які відносяться до підсистеми керування системою S_o. Задамо її так:

$$M_{V_{S_{0,1}}} = \begin{pmatrix} p(s_{1,0,1}) & \cdots & p(s_{1,0,n}) \\ \vdots & \ddots & \vdots \\ p(s_{12,0,1}) & \cdots & p(s_{12,0,n}) \end{pmatrix}, \quad (10)$$

де n – кількість компонент підсистеми S.

Таким чином, впливи на систему можуть бути ззовні неї та із середини. Але в сформованій множині впливів W містяться впливи, які можуть бути вторинними по відношенню до початкових впливів, тобто впливи які виникають після впливу на систему S_o. Також, цими вторинними впливами можуть бути і початкові впливи, тобто впливи початкові і вторинні можуть збігатись. Важливими для розгляду є руйнуючі та неруйнуючі впливи, які можуть впливати на систему, зокрема і цілеспрямовано для її деградації. Крім того, впливи можуть зловмисними або такими, що породжені виконанням завдань користувача чи системи S_o. Подальша деталізація компонентів та елементів системи, її архітектури потребуватиме виокремлення підмножин впливів з певними особливостями і їх врахування для забезпечення стійкості системи в цілому та її компонентів.

Впливи в комп'ютерних системах можуть відбуватись без подальших збурень системи S_o, тобто можуть відбуватись невидимо для давачів системи або відбуватись так, що будуть невидимими для неї. Але більшість впливів буде видима для системи. І завданням з розробки архітектури системи S_o є забезпечення її такими засобами, щоб вона могла бачити максимальну кількість подій в комп'ютерній системі та мережі.

Результатами дій впливів на систему S_o будуть наслідки. Вони можуть стати новими впливами або сформують в системі певний ланцюг дій. Впливи змінюватимуть стани системи. Наслідки впливів можуть активувати засоби системи S_o з активізації надання приманок чи пасток в комп'ютерній мережі або будуть мати відношення до дій системи для обробки подій. Але наслідки можуть породжувати нові впливи, які переважно впливатимуть на гнучку зміну та динамічну перебудову архітектуру системи S_o. Наслідки впливів можуть поділені на критичні та некритичні. Наслідки для системи S, також, можуть виникнути не в результаті відбування в комп'ютерній системі чи мережі певних процесів, які формуватимуть впливи, але і в результаті певного часу функціонування системи S_o та будуть результатом виконання певних дій в системі. Тобто, з'являтимуться в результаті функціонування системи S_o будуть для системи S_o вторинними. Враховуючи важливість наслідків і їх різноманітність для функціонування системи S_o введемо для них множину $M_{S,N}$, яку задамо її елементами так:

$$M_{S,N} = \left\{m_{S,n_1}, m_{S,n_2}, \dots, m_{S,n_{k_N}}\right\}, \quad (11)$$

де k_N – кількість наслідків; m_{S,n_l} – елемент множини, який кодує l – ий наслідок; $l = 1,2, \dots, k_N$.

Якщо врахувати, що наслідки впливів виникають в компонентах системи S_o в комп'ютерних системах, тоді сформуємо матрицю наслідків з врахуванням не системи в цілому, а саме її компонентів. Прояви наслідків задамо предикатами та окремо ймовірнісними функціями. Задання функціями із значеннями з проміжку $[0; 1]$ необхідно для використання в системі S_o для оціночного обчислення на певних кроках її функціонування. Якщо ж використовувати предикативне значення, то воно потрібно для встановлення факту наслідку і не завжди може бути чітким. Задамо матрицю наслідків з врахуванням компонентів системи згідно їх предикативних значень так:

$$M_{S,N,p} = \begin{pmatrix} p(m_{S,n_1,1,1}) & \cdots & p(m_{S,n_1,n,1}) \\ \vdots & \ddots & \vdots \\ p(m_{S,n_1,1,k_N}) & \cdots & p(m_{S,n_1,n,k_N}) \end{pmatrix}, \quad (12)$$

де n – кількість компонент підсистеми S; k_N – кількість наслідків.

Матрицю наслідків з врахуванням компонентів системи згідно їх ймовірнісних значень з проміжку $[0; 1]$ задамо так:

$$M_{S,N,\bar{\bar{f}}} = \begin{pmatrix} \bar{\bar{f}}(m_{S,n_1,1,1}) & \cdots & \bar{\bar{f}}(m_{S,n_1,n,1}) \\ \vdots & \ddots & \vdots \\ \bar{\bar{f}}(m_{S,n_1,1,k_N}) & \cdots & \bar{\bar{f}}(m_{S,n_1,n,k_N}) \end{pmatrix}, \quad (13)$$

де n – кількість компонент підсистеми S; k_N – кількість наслідків; $\bar{\bar{f}}$ – функція, яка обчислює ймовірність прояву наслідку і її значення знаходиться в проміжку $[0; 1]$.

Задані подання впливів та їх наслідків необхідні системі S_o для прийняття рішення щодо подальших дій. Без впливів із зовнішнього середовища система S_o буде перебувати в поточному стані і за появи

внутрішніх впливів оброблятиме їх. Система S_o під впливами та наслідками, які можуть породжувати знову впливи, буде змінювати стани, в яких вона перебуватиме у вузлах комп'ютерної мережі. Ці зміни можуть відбуватись синхронно в декількох компонентах одночасно або асинхронно в різних компонентах. При цьому впливи можуть збігатись в різних компонентах. Система S_o в своїх компонентах має різні стани. Її загальний стан аналізується підсистемою керування для визначення подальших кроків. До таких станів компонент системи S_o віднесемо активність / не активність функційних елементів підсистем компонент.

Нехай множина елементів підсистеми S задана множиною елементів так:

$$M_{S_i,p_e} = \{s_{i,0}, s_{i,1}, \dots, s_{i,N_S-1}\}, \tag{14}$$

де N_S – кількість елементів підсистем в компонентах підсистеми S; $i = 1,2,\dots,n$; n – кількість компонентів в підсистемі S.

Кількість функційних елементів кожної з підсистем компоненти всієї підсистеми S є не однаковою. Тому, для кожної з підсистем компоненти згідно формули (14) задамо вектори, компонентами яких будуть значення предикатів, які визначатимуть активність / не активність функційних елементів і позначатимуться відповідно значеннями {1} або {0}. Тоді, з врахуванням функційних елементів в підсистемах компонентів вектори їх станів в компонентах задамо так:

$$s_{i,j} = \left(p(s_{i,j,1}), p(s_{i,j,2}), \dots, p(s_{i,j,N_{s_{i,j}}})\right), \tag{15}$$

де $s_{i,j}$ - j-а підсистема i-ої компоненти; $s_{i,j,l}$ - функційний елемент j-ї підсистеми i-ої компоненти; $l = 1,2,\dots,N_{s_{i,j}}$; $N_{s_{i,j}}$ – кількість функційних елементів j-ї підсистеми i-ої компоненти; $i = 1,2,\dots,n$; n – кількість компонент системи S; $J = 0,1,2,\dots,N_{s_i} - 1$; N_{s_i} – кількість підсистем в компоненті S_i.

Кількість функційних елементів підсистем компонентів різна, тому вектори матимуть різну кількість своїх компонентів і не можуть бути зведеними в єдину матрицю. Значення для векторів $s_{i,j}$ можуть бути обчислені двома способами. Перший спосіб полягає в тому, що якщо активний хоча б один функційний елемент, тоді значення вектору $p(s_{i,j}) = 1$, інакше $p(s_{i,j}) = 0$ при $i = 1,2,\dots,n$, n – кількість компонент системи S, $J = 0,1,2,\dots,N_{s_i} - 1$, N_{s_i} – кількість підсистем в компоненті S_i. Тобто, отримуємо предикативне значення. Але може бути потрібно отримувати значення порівнювані із значеннями решти підсистем, тоді застосуємо другий спосіб, який базується на визначенні кількості активних функційних елементів щодо усіх елементів і обчислюється так:

$$\bar{g}(s_{i,j}) = \frac{\sum_{l=1}^{N_{s_{i,j}}} p(s_{i,j,l})}{N_{s_{i,j}}}, \tag{16}$$

де $p(s_{i,j,l})$ – значення предикату, яке вказує на активність / не активність функційного елементу і визначається числами {1} або {0} відповідно; $s_{i,j}$ - j-а підсистема i-ої компоненти; $s_{i,j,l}$ - функційний елемент j-ї підсистеми i-ої компоненти; $l = 1,2,\dots,N_{s_{i,j}}$; $N_{s_{i,j}}$ – кількість функційних елементів j-ї підсистеми i-ої компоненти; $i = 1,2,\dots,n$; n – кількість компонент підсистеми S; $J = 0,1,2,\dots,N_{s_i} - 1$; N_{s_i} – кількість підсистем в компоненті S_i.

Отримані значення для векторів $s_{i,j}$ в процесі функціонування системи S_o використовуватимуться як аргументи для визначення її подальших кроків і, тому, потребують унормування для врахування частки впливу. Здійснимо унормування їх предикативних значень так:

$$p_1(s_{i,j,l}) = \frac{p(s_{i,j,l})}{\sum_{l=1}^{N_{s_{i,j}}} p(s_{i,j,l})}, \tag{17}$$

де $i = 1,2,\dots,n$, n – кількість компонент підсистеми S, $J = 0,1,2,\dots,N_{s_i} - 1$, $s_{i,j,l}$ - функційний елемент j-ї підсистеми i-ої компоненти; $l = 1,2,\dots,N_{s_{i,j}}$; $N_{s_{i,j}}$ – кількість функційних елементів j-ї підсистеми i-ої компоненти.

Тоді, справедливе співвідношення: $\sum_{l=1}^{N_{s_{i,j}}} p_1(s_{i,j,l}) = 1$. Так унормовані значення можуть бути аргументами в формулі (16):

$$\bar{g}(s_{i,j}) = \frac{\sum_{l=1}^{N_{s_{i,j}}} p_1(s_{i,j,l})}{N_{s_{i,j}}}, \tag{18}$$

де $p_1(s_{i,j,l})$ – унормоване значення предикату $p(s_{i,j,l})$; $s_{i,j}$ - j-а підсистема i-ої компоненти; $s_{i,j,l}$ - функційний елемент j-ї підсистеми i-ої компоненти; $l = 1,2,\dots,N_{s_{i,j}}$; $N_{s_{i,j}}$ – кількість функційних елементів j-ї підсистеми

i-ої компоненти; $i = 1,2,\dots,n$; n – кількість компонент підсистеми S; $J = 0,1,2,\dots,N_{s_i} - 1$; N_{s_i} – кількість підсистем в компоненті S_i.

Формули (16) - (18) визначають значення безпосередньо для компонент підсистеми S. Тобто, всі результати обчислень стосуються окремих компонент. Але важливими, також, є значення для однакових підсистем компонент. Підсистеми компонент містять однакові функційні елементи. Хоча не всі компоненти можуть містити всі підсистеми, особливо це стосується компонент, в яких міститься підсистема керування системою S_o та функційні елементи для роботи певних визначених приманок. Це потрібно для оцінювання функціонування системи S_o в розрізі її підсистем першочергово, а не компонент. Оскільки в переважній більшості компонент містяться однакові підсистеми, то дослідження процесів, які в них відбуваються, доповнить базу аргументів для визначення подальших дій системи S_o. Задамо вектори для кожної з підсистем у всіх компонентах підсистеми S. Якщо відповідна підсистема буде відсутньою у компоненті, тоді у відповідній компоненті вектору її значення встановимо $\{0\}$. Для обчислень в розрізі підсистем враховуватимемо тільки компоненти, в яких наявні підсистеми. Це важливо, бо система може мати таку архітектуру, що значна частина компонент не міститиме певної підсистеми, і обчислення для всіх компонентів спотворюватиме точність результату.

Тому, визначимо значення для підсистем з врахуванням їх задання формулами (14) і (15) так:

$$p_{2,j}(s_{i,j},j) = \begin{cases} 1,\ \text{при всіх } i = 1,2,\dots,n \text{ та існує хоча б одна} \\ \text{активна активна } j-\text{а підсистема хоча б в одній} \\ \text{зі всіх компонент системи } S, n-\text{кількість} \\ \text{компонент системи, інакше} \\ 0, \text{при всіх } i = 1,2,\dots,n \text{ та не існує жодної активної} \\ j-\text{ї підсистеми хоча б в одній зі всіх компонент} \\ \text{системи } S, n-\text{ кількість компонент системи } S, \\ J = 0,1,2,\dots,N_{s_i}-1, N_{s_i}-\text{ кількість підсистем} \\ \text{в компоненті } S_i. \end{cases}, \tag{19}$$

Тобто, отримуємо предикативне значення для підсистеми, яка міститься в більшості компонент або в усіх компонентах системи. Також, потрібні значення підсистем, які обчислюються за кількістю активних функційних елементів щодо усіх елементів так:

$$\overline{g_{2,j}}(s_{i,j},j) = \frac{\sum_{i=1}^{n}\sum_{l=1}^{N_{s_{i,j}}} p(s_{i,j,l})}{(n-k)N_{s_{i,j}}}, \tag{20}$$

де $p(s_{i,j,l})$ – значення предикату, яке вказує на активність / не активність функційного елементу і визначається числами $\{1\}$ або $\{0\}$ відповідно; $s_{i,j}$ - j-а підсистема i-ої компоненти; $s_{i,j,l}$ - функційний елемент j-ї підсистеми i-ої компоненти; $l = 1,2,\dots,N_{s_{i,j}}$; $N_{s_{i,j}}$ – кількість функційних елементів j-ї підсистеми i-ої компоненти; $i = 1,2,\dots,n$; n – кількість компонент системи S; $J = 1,2,\dots,N_{s_i}$; N_{s_i} – кількість підсистем в компоненті S_i; k – кількість компонент в системі, в яких відсутня J - а підсистема.

Отримані значення для підсистем в процесі функціонування підсистеми S використовуватимуться як аргументи для визначення її подальших кроків і, тому, потребують унормування для врахування частки впливу. Здійснимо унормування їх предикативних значень так:

$$p_{3,j}(s_{i,j,l},j) = \frac{\sum_{l=1}^{N_{s_{i,j}}} p(s_{i,j,l})}{\sum_{i=1}^{n}\sum_{l=1}^{N_{s_{i,j}}} p(s_{i,j,l})}, \tag{21}$$

де $p(s_{i,j,l})$ – значення предикату, яке вказує на активність / не активність функційного елементу і визначається числами $\{1\}$ або $\{0\}$ відповідно; $s_{i,j}$ - j-а підсистема i-ої компоненти; $s_{i,j,l}$ - функційний елемент j-ї підсистеми i-ої компоненти; $l = 1,2,\dots,N_{s_{i,j}}$; $N_{s_{i,j}}$ – кількість функційних елементів j-ї підсистеми i-ої компоненти; $i = 1,2,\dots,n$; n – кількість компонент системи S; $J = 0,1,2,\dots,N_{s_i} - 1$; N_{s_i} – кількість підсистем в компоненті S_i.

Тоді, справедливе співвідношення: $\sum_{l=1}^{N_{s_{i,j}}} p_{3,j}(s_{i,j,l},j) = 1$.

Так унормовані значення можуть бути аргументами в формулі (20):

$$\overline{g_{3,j}}(s_{i,j},j) = \frac{\sum_{l=1}^{N_{s_{i,j}}} p_{3,j}(s_{i,j,l},j)}{(n-k)N_{s_{i,j}}}, \tag{22}$$

де $p_1(s_{i,j,l})$ – унормоване значення предикату $p(s_{i,j,l})$; $s_{i,j}$ - j-а підсистема i-ої компоненти; $s_{i,j,l}$ - функційний елемент j-ї підсистеми i-ої компоненти; $l = 1,2,\dots,N_{s_{i,j}}$; $N_{s_{i,j}}$ – кількість функційних елементів j-ї підсистеми

i-ої компоненти; $i = 1,2, \dots, n$; n – кількість компонент підсистеми S; $J = 0,1,2, \dots, N_{s_i} - 1$; N_{s_i} – кількість підсистем в компоненті S_i; k – кількість компонент в системі, в яких відсутня J - а підсистема.

Формули (21) і (22) дозволяють обчислювати значення безпосередньо для підсистем компонент системи S_o. Тобто, всі результати обчислень стосуються окремих підсистем компонент.

Отримані таким чином числові характеристики елементів та компонентів системи S_o дозволяють здійснювати її оцінювання поточного стану для визначення її подальших кроків. Вони є частиною загальної сукупності характеристик і основною частиною для керування підсистемою S. Активність та не активність функційних елементів підсистем компонентів підсистеми S можна задати станами, в яких відображатиметься поточний стан системи S. Частина станів системи може мати зв'язки між собою. Зміна станів функційних елементів може відбуватись динамічно. Впливи на систему S будуть активувати функційні елементи і, відповідно, змінюватимуть їх стани. За результатами опрацьовування впливів будуть наслідки, за якими можуть змінюватись стани певних функційних елементів. Враховуючи, що система S має розподілену архітектуру, то її компоненти можуть перебувати в однакових станах в певні моменти часу. При залученні засобів проти ЗПЗ та КА стани відповідних спеціалізованих функційних елементів теж можуть збігатись. Перебуваючи в активному стані певні функційні елементи можуть активізувати інші функційні елементи, тобто змінити їх стан. Все це впливатиме на прийняття рішень щодо подальших кроків підсистемою керування підсистемою S і повинно нею враховуватись. Тому, необхідним є розроблення відповідних методів, які б враховували такі особливості в архітектурі системи керування в розрізі завдань, які до неї висуваються з самостійного визначення подальших кроків системи S. В зв'язку з такими вимогами архітектура підсистеми керування підсистемою S матиме особливості та автономність в загальній архітектурі системи S. Таким чином, функційні елементи та їх стани, а також, зокрема, функційні елементами підсистеми керування зі своїми станами, будуть мати однакові способи переходу зі стану в стан, але наслідки з цих переходів хоча, і породжуватимуться ними та впливатимуть в прямому і зворотному напрямах, все-таки функційні елементи підсистеми керування матимуть вищий пріоритет щодо породження впливів на решту функційних елементів.

Модель функціонування обманних систем згідно популяційних алгоритмів

В процесі свого функціонування обманні системи згідно моделі їх задання (формули (3) та (4)) повинні самостійно обирати свої наступні кроки. Особливо це актуально в момент здійснення КА на ресурси та засоби корпоративних мереж. Варіантів таких кроків, як правило, є багато. В момент здійснення КА система може мати багато варіантів для відповіді на них. Також, під час штатного функціонування вона повинна реагувати на зміни в кількості її компонент, бо частина комп'ютерних станцій може вимикатись та вмикатись тим самим змінюючи склад системи. Приманки та пастки, які є частиною системи можуть виконувати певні завдання, або не виконувати, або частина з них буде виконувати і решта не буде виконувати. Тому, варіантів для вибору в таких системах багато. Вони не можуть здійснювати повний перебір таких варіантів, оскільки стан вузлів корпоративних мереж стрімко змінюється і актуальність частини варіантів для прийняття їх в якості можливого наступного кроку системи швидко втрачається. В зв'язку з цим, а також з тим, що відповіді та поведінка системи в момент КА повинна бути різною, необхідно в архітектурі обманних систем реалізувати модель такої поведінки, яка б дозволила уникнути повного перебору варіантів, не вела б до деградації та розбалансування дій, забезпечувала б ефективне функціонування, складність розуміння зловмисниками, не була б збіжною за невелику кількість кроків та могла б підтримувати дії системи щодо певних впливів напротязі часу здійснення таких впливів. Таким чином, основними вимогами до організації функціонування обманних систем є забезпечення сталої поведінки під час тривалих КА та під час тривалої експлуатації.

При виборі з потенційних варіантів подальших кроків обманних систем необхідно розв'язувати задачу оптимізації, в якій множина розв'язків є дискретною чи бути зведеною до дискретної і є варіантами кроків. Така задача є задачею комбінаторної оптимізації і, при цьому, обраними мають бути не окремі одиничні наступні кроки системи, а послідовності кроків. В цих послідовностях кроків можуть змінюватись наступні кроки, але перші кроки повинні пов'язувати декілька наступних кроків. Так має бути сформовано послідовність кроків. Це доповнює вимоги та обмеження для таких задач комбінаторної оптимізації.

Оскільки зловмисники при проведенні КА використовують розроблені моделі поведінки, які все-таки переважно відповідають їх характерним рисам поведінки або типовим стратегіям поведінки зловмисників, то для забезпечення вимоги щодо підтримання тривалої роботи обманних систем з приманками і пастками в момент здійснення КА моделі поведінки обманних систем на противагу моделям КА повинні містити алгоритми оптимізації, які відображають природню поведінку. Тоді, зловмисникам будуть протидіяти обманні системи, які функціонуватимуть згідно поведінкових моделей.

Для розроблення такої поведінкової моделі, щоб забезпечити функціонування обманних систем з врахуванням того, що на кожному кроці потрібно буде обирати один варіант з багатьох, використаємо алгоритм, який обирають для вирішення задач, коли точне рішення невідоме або його отримання є занадто складним і затратним за часом. Він заснований на певному оцінюванні та досвіді, який формується з часом для того, щоб знайти прийнятне рішення у більшості випадків, хоча і не обов'язково оптимальне. Алгоритм

для моделі виберемо з множини популяційних алгоритмів, бо вони характеризуються тим, що на кожному наступному кроці опрацьовують декілька розв'язків одночасно, а результати пошуку та вибору розв'язків формують та розширюють досвід, зокрема і колективний досвід для агентів. Взагалі популяційний алгоритм буде відображати роботу багатьох агентів в системі. Клас з множини популяційних алгоритмів для використання їх в поведінковій моделі обманних систем, враховуючи специфіку функціонування та завдань, візьмемо з алгоритмів, які натхненні живою природою. На основі такого вибору задамо концептуальну модель обманних систем з приманками та пастками згідно формул (3) та (4) і вимоги щодо формування поведінки згідно популяційних алгоритмів так:

$$M_{S_o} = \left(S_o, A_{o,p}, G_{o,p}\right), \tag{23}$$

де S_o – обманна система, яку задано елементами множини за формулами (2.8) та (2.9); $A_{o,p}$ – множина популяційних алгоритмів, елементи якої означують функційний елемент $s_{i,0,6}$ (рис. 5) підсистеми керування для визначення подальших кроків у функціонуванні системи; $G_{o,p}$ – граф зв'язків між елементами множини S_o та одним з елементів множини $A_{o,p}$.

Згідно концептуальної моделі M_{S_o} (формула (3)) наявність в системі елементу з множини популяційних алгоритмів $A_{o,p}$ є обов'язковим. Тоді, зображення графу зв'язків $G_{o,p}$ між елементами множини S_o та одним з елементів множини $A_{o,p}$ задамо на рис. 6.

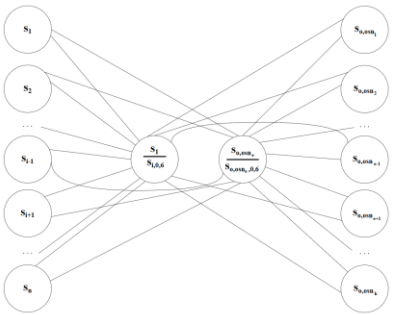

Рис. 6. Граф зв'язків $G_{o,p}$

Граф зв'язків $G_{o,p}$ є підграфом повного графу, в якому вершини, що відображають елементи множин основної частини обманних систем без приманок і пасток $S_{o,osn}$ і частини обманних систем саме з приманками і пастками S, пов'язані між собою. На рис. 6 такі позначення: n – це кількість приманок і пасток в системі; $n \geq 2$, S_i - i-та приманка чи пастка; k – це кількість компонентів та модульних елементів основної частини обманної системи; $k \geq 1$, S_{o,osn_j} - j-тий компонент або модульний елемент; $j = 1,2, \dots, k$. Граф зв'язків $G_{o,p}$ формується динамічно в системі в залежності від компоненти, в якій будуть прийматись рішення щодо подальших кроків згідно популяційного алгоритму. При зміні компоненти граф зв'язків $G_{o,p}$ буде змінено автоматично обманною системою.

Функційний елемент $s_{i,0,6}$ з підсистеми S має такого ж відповідника $s^*_{o,osn_l,0/6}$ з підсистеми $S_{o,osn}$. Центр системи означується наявністю саме одним з цих елементів.

Таким чином, особливістю розробленої концептуальної моделі обманних систем з приманками і пастками в корпоративних мережах є поділ компонентів на компоненти підсистеми, якою забезпечуються обманні технології, та підсистеми безпосередньо з приманками і пастками, а також наявністю в них підсистеми прийняття рішень на основі популяційних алгоритмів для заплутування зловмисників та підтримування стійкої відповіді на цілеспрямовані тривалі зловмисні дії в процесі здійснення двоцільових атак.

Розглянемо детальніше популяційні алгоритми в контексті їх використання в обманних системах для забезпечення підтримки тривалої поведінки системи на певну тривалу зловмисну дію чи КА. Такі алгоритми вирішують задачі оптимізації. Оптимізація стосується вибору наступних кроків системи з певної кількості можливих кроків для виконання завдань системи. Важливою особливістю при виборі має бути спроможність системи з використанням певного популяційного алгоритму не тільки виконати декілька наступних кроків, але підтримувати їх вибір тривалий час поки здійснюється певна КА чи дії ЗПЗ. Якщо є система вибиратиме кроки і через певний час такий вибір вичерпується, тобто вона починає деградувати, то це не дозволяє виконати належну протидію КА та ЗПЗ і переважно пов'язано з передчасною збіжністю реалізованого в ній методу.

При виборі наступних кроків системи з потенційних кроків може виникнути ситуація, коли система обиратиме кожного разу з меншої кількості варіантів і врешті кількість кроків для вибору стане нульовою, а КА чи дії ЗПЗ будуть продовжуватись. Таким чином, виникне подія, суть якої в передчасній збіжності і вона буде відображати стагнацію методу оптимізації в локальному оптимумі. Це перешкоджає збіжності методу

до глобального оптимуму. Алгоритми, які базуються на популяціях, мають високу здатність уникати локального оптимуму, бо під час оптимізації вони опрацьовують набір рішень. Також, з реалізацією мультиагентних технологій в архітектурі обманних систем можливий обмін інформацією між компонентами та організація допомоги їм у подоланні проблем в пошукових просторах.

Популяційних алгоритмів відомо багато. Досить часто значна частина з них може не бути представлена деталізованою послідовністю кроків, як того вимагає означення та властивості алгоритмів, а є лише ілюстрацією ідеї чи дій, які натхненні природою, тобто не мають розроблених загальноприйнятих кроків за винятком загальної структури, за якою вони відносяться до популяційних алгоритмів, та відповідної загальноприйнятої ідентифікації.

В контексті розв'язуваної задачі щодо покращення функціонування обманних систем в частині забезпечення тривалого реагування на тривалу КА чи тривалі дії ЗПЗ, а також організації використання приманок і пасток, виникає модель відносин між діями та засобами зловмисників з однієї сторони та діями обманних систем з приманками і пастками з другої сторони. Суть цих дій полягає в тому, що КА та ЗПЗ спрямовані і переміщуються в напряму визначених зловмисниками об'єктів в корпоративних мережах, а назустріч їм для забезпечення безпеки та захисту корпоративних мереж та їх ресурсів спрямовані приманки і пастки, а також компоненти і елементи обманних систем, ховаючи за собою реальні об'єкти. Результатом контактів між засобами проведення КА та ЗПЗ з однієї сторони та приманками і пастками обманних систем будуть такі варіанти:

1) КА чи ЗПЗ завершились в приманках чи пастках;

2) КА чи ЗПЗ потрапили в приманки чи пастки, але при цьому їх активність продовжилась на інших напрямах;

3) КА чи ЗПЗ надали змогу зловмисникам отримати контроль над приманками чи пастками, з якими увійшли в контакт.

КА чи ЗПЗ можуть також обійти приманки чи пастки. А також, при цьому, продовжити свої дії для досягнення мети.

Обманні системи у варіанті з втратою приманок і пасток за певними ознаками можуть побачити таку втрату приманок і пасток та повинні використати захисні засоби для уникнення їх використання зловмисником. Тому, приманки і пастки потрібно застосовувати в складі обманних систем, які можуть забезпечити контроль їх застосування. Якщо є обманні системи не помітили втрати контролю над приманками і пастками, тоді рівень безпеки корпоративних мереж може бути знижений через впливи зловмисників. Для уникнення таких подій необхідні засоби в обманних системах, що відповідатимуть за контроль за приманками і пастками. Інакше, зловмисник успішно класифікує хибні об'єкти для атак, що відповідає двоцільовій моделі КА.

Назустріч КА та ЗПЗ можуть бути спрямовані декілька приманок і пасток. А також, КА і ЗПЗ можуть мати декілька цілей в корпоративних мережах, зокрема, вони можуть проводитись одночасно і, при цьому, бути різними. Тоді, обманні системи повинні забезпечувати всі наявні вектори КА та дій ЗПЗ відповідними хибними об'єктами для атак. Зловмисники теж можуть використовувати хибні КА чи організовувати хибні впливи ЗПЗ для відволікання ресурсів і часу систем безпеки та захисту корпоративних мереж. Обманні системи повинні спрямовувати КА та ЗПЗ на хибні об'єкти для атак. Крім того, вибір наступних кроків обманних систем і, зокрема, кроків для активізації та наближення приманок і пасток, передбачає вирішення завдання оптимізації. Якщо КА та ЗПЗ завершились як активності в приманках та пастках, повністю або частково, то така подія буде відповідати алгоритму молі і вогню. Таким чином, вибір приманок та пасток для КА та дій ЗПЗ є кроками системи, які потребують оптимізації.

Алгоритм молі і вогню відноситься до популяційних алгоритмів та вирішує завдання оптимізації і належить до класу алгоритмів натхненних природою. Суть натхнення природою для цього алгоритму полягає в способі руху молі в природі. Загально прийнято цей рух називають поперечною орієнтацією. Моль літає вночі. При побудові маршруту вона зберігає фіксований кут по відношенню до Місяця, завдяки чому може долати великі відстані по прямій лінії. За другим випадком вона попадає в пастку на спіральній доріжці навколо штучного освітлення. Ці два випадки є основою для досягнення оптимізації при розв'язуванні задач з використанням алгоритму молі і вогню. Для застосування алгоритму введемо позначення об'єктів корпоративних мереж та встановимо зв'язки між ними.

Задамо вузли корпоративних мереж множиною $M_{kn,v} = \left\{m_{kn,v,1}, m_{kn,v,2}, \dots, m_{kn,v,N_{kn,v}}\right\}$, де $N_{kn,v}$ – кількість вузлів корпоративних мереж. Здійснимо їх поділ на підмножини за ознакою належності до певного сегменту корпоративних мереж, тоді

$$M_{kn,p} = \bigcup_{q=1}^{N_{kn,v,p}} M_{kn,v,p,q}, \tag{24}$$

де $M_{kn,v,p,q}$ – підмножини множини $M_{kn,p}$; $N_{kn,v,p}$ – кількість підмножин вузлів корпоративних мереж в множині $M_{kn,p}$; $q = 1,2, \dots, N_{kn,v,p}$.

Задамо підмножини $M_{kn,v,p,q}$ $(q = 1,2,\dots,N_{kn,v,p}$, $N_{kn,v,p}$ – кількість підмножин вузлів корпоративних мереж в множині $M_{kn,p})$ з формули (2.29) матрицею $M_{kn,v,p,z}$, в якій елементами рядків будуть вузли корпоративних мереж, що належатимуть до одного сегменту, так:

$$M_{kn,v,p,z} = \begin{pmatrix} m_{kn,v,p,z,1,1} & \cdots & m_{kn,v,p,z,1,N_{kn,v,p,z}} \\ \vdots & \ddots & \vdots \\ m_{kn,v,p,z,N_{kn,v,p}} & \cdots & m_{kn,v,p,z,N_{kn,v,p},N_{kn,v,p,z}} \end{pmatrix}, \quad (25)$$

де $m_{kn,v,p,z,i,j}$ – елемент підмножини $M_{kn,v,p,i}$ $(i = 1,2,\dots,N_{kn,v,p}$, $N_{kn,v,p}$ – кількість підмножин вузлів корпоративних мереж в множині $M_{kn,p})$; $N_{kn,v,p,z}$ – максимальна кількість вузлів в сегментах мережі включно з маршрутизатором; j – номер стовпця матриці; $j = 1,2,\dots,N_{kn,v,p,z,N_{kn,v,p,z,j}}$; $j = 1,2,\dots,N_{kn,v,p,z,j}$; $N_{kn,v,p,z,j}$ - кількість вузлів в j-му сегменті мережі включно з маршрутизатором; $N_{kn,v,p,z,j} \le N_{kn,v,p,z}$; $i = 1,2,\dots,N_{kn,v,p}$; $N_{kn,v,p}$ – кількість підмножин вузлів корпоративних мереж в множині $M_{kn,p}$.

Якщо $N_{kn,v,p,z,j} < N_{kn,v,p,z}$ для частини підмножини, то всі елементи матриці $M_{kn,v,p,z}$ від $N_{kn,v,p,z,j} + 1$ по $N_{kn,v,p,z}$ для всіх рядків, в яких виконується це співвідношення, встановлюємо таким, що дорівнює $p_{M_{kn,v,p,z}}$.

В матриці $M_{kn,v,p,z}$ відображено групування вузлів корпоративних мереж за сегментами. Встановимо порядок між вузлами в сегментах корпоративних мереж та задамо їх матрицею векторів $M_{kn,v,p,z,vect}$ на основі матриці $M_{kn,v,p,z}$ (формула (2.30)) так:

$$M_{kn,v,p,z,vect} = \begin{pmatrix} m_{kn,v,p,z,vect,1,1} & \cdots & m_{kn,v,p,z,vect,1,N_{kn,v,p,z}} \\ \vdots & \ddots & \vdots \\ m_{kn,v,p,z,vect,N_{kn,v,p},1} & \cdots & m_{kn,v,p,z,vect,N_{kn,v,p},N_{kn,v,p,z}} \end{pmatrix}, \quad (26)$$

де $m_{kn,v,p,z,vect,i,j}$ – компонента вектору з елементів підмножини $M_{kn,v,p,i}$ $(i = 1,2,\dots,N_{kn,v,p}$, $N_{kn,v,p}$ – кількість підмножин вузлів корпоративних мереж в множині $M_{kn,p})$; $N_{kn,v,p,z}$ – максимальна кількість вузлів в сегментах мережі включно з маршрутизатором; j – номер стовпця матриці; $j = 1,2,\dots,N_{kn,v,p,z,N_{kn,v,p,z,j}}$; $j = 1,2,\dots,N_{kn,v,p,z,j}$; $N_{kn,v,p,z,j}$ - кількість вузлів в j-му сегменті мережі включно з маршрутизатором; $N_{kn,v,p,z,j} \le N_{kn,v,p,z}$; $i = 1,2,\dots,N_{kn,v,p}$; $N_{kn,v,p}$ – кількість підмножин вузлів корпоративних мереж в множині $M_{kn,p}$.

На рис. 7 зображено розміщення вузлів корпоративних мереж, які задано формулами (24) - (26).

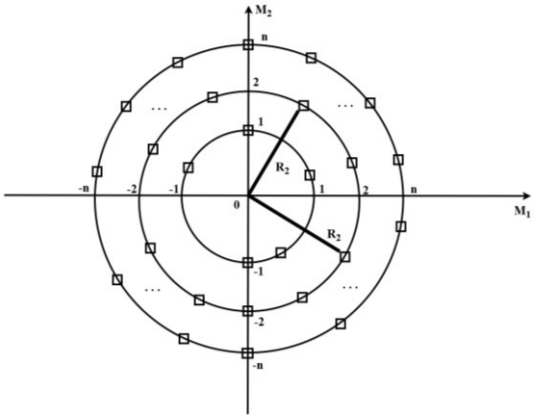

Рис. 7. Подання вузлів корпоративних мереж на координатній площині

Квадрати кожного кола відображають вузли корпоративних мереж, які належать одному сегменту. Перше коло відображає сегмент, в якому є засоби під'єднання до глобальних мереж, а також, його вузли можуть формувати демілітаризовану зону мережі. Чим далі кола знаходяться від центру координат тим вищий рівень безпеки у комп'ютерних станціях або тим довше за часом передається повідомлення від центру в точці О до відповідного кола порівняно з ближчими колами. Відстань до вузлів, які зображено квадратами, до центру в точці О є однаковою для всіх них, якщо вони знаходяться на лінії певного кола. Наприклад, R_2 – є радіусом другого кола і два відрізки відображають однакову відстань до двох вузлів в одному сегменті мережі.

Порядок вузлів, які розміщені на певних колах, може змінюватись обманними системами. Частина вузлів в процесі КА може вимикатись ними або блокуватись. Тоді, їх можна зображати різними кольорами на колах в залежності від доступності до них і їх активності.

В контексті обманних систем введемо множину впливів на корпоративні мережі, які можуть здійснювати зловмисники з використанням ЗПЗ та проведенням КА, і множину реакцій на впливи.

Задамо множину дій M_{vp} зловмисників, які спричиняють впливи на корпоративні мережі і реалізують КА та поширення ЗПЗ, згруповано за типами так:

$$M_{vp} = \begin{pmatrix} m_{vp,1} \\ m_{vp.2} \\ \dots \\ m_{vp,N_{vp}} \end{pmatrix}, \quad (27)$$

де $m_{vp.q}$ - q-а дія зловмисників, що реалізує вплив на корпоративні мережі для здійснення КА або поширення ЗПЗ, і характеризує певний тип таких дій; $q = 1,2, \dots, N_{vp}$; N_{vp} – кількість типів дій зловмисників.

Дії зловмисників, які задано множиною M_{vp} за формулою (27) характеризують певний тип і формують типовий клас. Цей типовий клас характеризує певний тип КА чи дії ЗПЗ. Перелік дій в самому класі може бути різним за кількістю етапів, послідовністю та залученими функціями, але він характеризує спорідненість їх та однакову спрямованість на корпоративні мережі. Такий поділ є необхідними для формування реакцій обманних систем, які можна задати окремою множиною реакцій та реалізувати як частину кроків для забезпечення безпеки та захисту корпоративних мереж. На певний типовий клас дій зловмисника як правило розробляють відповідні реакції систем. Тому, для обманних систем з приманками та пастками необхідно задати множину дій або реакцій на типові або конкретні дії зловмисників, які здійснюють через впливи на корпоративні мережі.

Введемо множину дій $M_{os,d}$ обманних систем з приманками і пастками, зокрема і тих, які є реакціями на зловмисні впливи на корпоративні мережі, так:

$$M_{os,d} = \begin{pmatrix} m_{os,d,1} \\ m_{os,d.2} \\ \dots \\ m_{os,d,N_{os,d}} \end{pmatrix}, \quad (28)$$

де $m_{os,d.q}$ - q-а дія обманних систем з приманками і пастками, що реалізує також реакції на зловмисні впливи на корпоративні мережі для здійснення КА або поширення ЗПЗ, і характеризує певний тип таких дій; $q = 1,2, \dots, N_{os,d}$; $N_{os,d}$ – кількість дій обманних систем з приманками і пастками.

Множини дій зловмисників M_{vp} та дій обманних систем з приманками і пастками $M_{os,d}$, які визначені за формулами (27) і (28), є детермінованими щодо використовуваних команд. Кількість команд може з часом бути збільшена, але їх кількість скінчена і залежить від наявних операційних систем, систем програмування та апаратно-програмних засобів, які використовуються зловмисниками та користувачами корпоративних мереж. Тому, множини дій зловмисників та обманних систем з приманками і пастками мають є скінченними, але кожна з дій може бути подана різною кількістю допустимих в мережах та системах команд. Таким чином, множину дій $M_{os,d}$ обманних систем з приманками і пастками розділимо на такі підмножини:

1) $M_{os,d,1}$ – підмножина дій, які забезпечують функціонування безпосередньо обманних систем без приманок і пасток;

2) $M_{os,d,2}$ – підмножина дій, які забезпечують функціонування обманних систем винятково в частині керування приманками і пастками;

3) $M_{os,d,3}$ – підмножина дій, які забезпечують функціонування безпосередньо обманних систем без приманок і пасток в частині реакції на дії та впливи в корпоративних мережах включно з штатними та із зловмисними діями;

4) $M_{os,d,4}$ – підмножина дій, які забезпечують функціонування обманних систем для залучення приманок і пасток з метою реагування на впливи та дії в корпоративних мережах;

5) $M_{os,d,5}$ – підмножина дій, які забезпечують функціонування винятково приманок і пасток без залучення керування обманними системами.

Підмножини дій $M_{os,d,3}$ та $M_{os,d,4}$ відносяться до таких, що відповідають за опрацювання множини дій M_{vp} зловмисників. Вони спрямовані на здійснення класифікації подій в корпоративних мережах, які можуть бути віднесені до КА чи дій ЗПЗ.

В підмножинах дій $M_{os,d,1}$ та $M_{os,d,2}$ наявні елементи, які забезпечують адаптивність та перебудову обманних систем включно з керуванням приманками і пастками за наявності необхідних передумов. Частина дій з цих підмножин може поєднуватись з діями підмножин $M_{os,d,3}$ та $M_{os,d,4}$. Особливо коли для здійснення класифікації подій в корпоративних мережах в процесі здійснення КА чи поширення ЗПЗ зловмисниками необхідно здійснити зміну архітектури основної частини обманних систем, активізувати частину приманко і пасток тощо. Тобто, крім функціоналу з виявлення КА та дій ЗПЗ, який забезпечується діями підмножин $M_{os,d,3}$ та/або $M_{os,d,4}$ потрібно забезпечити виконання дій з підмножин $M_{os,d,1}$ та/або $M_{os,d,2}$.

Всі ці підмножини дають змогу формувати нові послідовності кроків обманних систем, тобто є основою для формування послідовності кроків обманних систем при тривалих атаках. Безпосередньо підмножини множини дій $M_{os,d}$ обманних систем з приманками і пастками містять конкретні послідовності кроків для виявлення та/чи класифікації КА та дій ЗПЗ, але при цьому в послідовностях цих кроків можуть бути засоби, які не завжди є чітко детермінованим, а можуть містити кроки, в яких задано ітераційні процеси тощо. Але такі кроки будемо розглядати одиничними кроками з урахуванням того, що при їх виконанні не

повинно бути переривання та переключення на інші кроки, тобто вони повинні отримувати всі необхідні ресурси комп'ютерних систем та виконуватись за один такт. В такому позиціонуванні вони будуть розглядатись як окремі кроки із послідовності загальних кроків, які повинна визначити для себе кожна обманна система в залежності від поточних подій і стану корпоративних мереж. Таким чином, задамо послідовність кроків обманних систем з приманками і пастками такими векторами:

$$v_{os,i} = \left(v_{os,i,1}, v_{os,i,2}, \dots, v_{os,i,N_{os,i}}\right), \tag{29}$$

де $v_{os,i,j}$ – крок обманної системи з приманками і пастками; $j = 1,2, \dots, N_{os,i}$; $N_{os,i}$ – кількість кроків обманних систем з приманками і пастками для вирішення певного завдання, які визначено системою в поточний момент часу; i – i-та послідовність кроків для вирішення певного завдання системи.

Кроки обманних систем можуть обиратись з врахування попередньої історії з опрацювання повторюваних подій або частково використовувати такі відомості. Але в архітектуру обманних систем закладено послідовності кроків для вирішення типових відомих КА та дій ЗПЗ. Крім того, такі обманні системи повинні мати засоби для модифікації своїх наступних кроків на основі використання базових кроків. До таких кроків відносяться кроки із керування та залучення до процесу виявлення КА та дій ЗПЗ приманок та пасток. Враховуючи специфіку обманних систем, то підбір кроків повинен бути таким, щоб уникати повторюваності дій, інакше зловмисники зможуть вивчати поведінку засобів та систем захисту корпоративних мереж. Тобто, реакції обманних систем на впливи, причому на певні конкретні впливи, можуть бути різними, тому і послідовності кроків, які визначатимуться такими системами будуть різними.

Серед кроків обманних систем з приманками і пастками важливими є такі, що впливають на активність та розміщення приманок і пасток, безпосередньо під час впливів на корпоративні мережі. При виборі таких послідовностей кроків виникають проблеми, які пов'язані із здійсненням повного перебору варіантів та вибору одного з них. Але така стратегія не є ефективною, оскільки варіантів послідовностей кроків дуже багато і повний перебір всіх варіантів буде затратним по часу. Крім того, не має гарантії, що обраний варіант в поточний момент коли він буде виконуватись не втратив свою актуальність в швидко змінюваному середовищі корпоративних мереж. Тому, вибір такого варіанту може бути здійснено з певного локального набору кроків. Основною вимогою для такого вибору має бути критерій щодо часу та актуальності підібраного вектору послідовності кроків. Крім того, обрана послідовність кроків повинна бути такою, щоб адекватно відповідати на виявлені впливи в корпоративних мережах від КА та дій ЗПЗ. Першочергово така відповідність стосується забезпечення тривалої відповіді на здійснювані впливи, тобто уникнення збіжності обраного варіанту послідовності кроків, коли впливи продовжуються, а кроки обманних систем вже вичерпались.

Підмножина кроків з векторів, які задають послідовності кроків обманних систем, може відноситись до кроків з керування приманками і пастками. Тоді, використання популяційного алгоритму, наприклад алгоритму молі і вогню, в архітектурі обманних систем з приманками і пастками може відноситись, наприклад, до прийняття рішень щодо розміщення приманок і пасток під час здійснення зловмисником КА чи виконанні дій ЗПЗ, які були встановлені давачами в корпоративних мережах.

Розглянемо спочатку конфігурування приманок і пасток для випадку, коли здійснюється КА на корпоративні мережі. Тоді, точку потенційного проникнення в корпоративну мережу можемо прийняти за точку О (0;0) на координатній площині (рис. 1), а комп'ютерні станції, які зображені на рис. 1, можна прийняти такими, що серед них будуть приманки і пастки окремими вузлами або поєднаними з реальними комп'ютерними станціями чи серверами. В такій інтерпретації алгоритму молі і вогню вплив від КА буде активувати рух приманок і пасток до нього для його нівелювання або фіксування в пастках. Традиційна схема алгоритму молі і вогню передбачає рух молі, який в даному випадку є відповідником КА, до джерела штучного освітлення, яке є відповідником приманці чи пастці. Тоді, другим випадком для конфігурування приманок і пасток є такий, при якому КА чи дії ЗПЗ відбувають у вузлах корпоративних мереж та зміщують в сторону приманки і пастки, які активовані обманною системою і позиціонують себе як ресурс, що є метою КА чи дій ЗПЗ.

Якщо розглядати дії ЗПЗ, включно з поширенням між вузлами корпоративних мереж, то вузол, в якому активізується ЗПЗ, на рис. 1 можна вважати таким, що представляє точку О. Тоді, решта вузлів будуть формувати оточення цього вузла і в частині з них будуть приманки і пастки окремо, а в частині будуть приманки і пастки разом з компонентами реальних інформаційних систем. При активізації ЗПЗ приманки і пастки будуть активізовані обманними системами та будуть себе проявляти в напряму руху до точки О. Або може бути застосований зворотній погляд на використання алгоритму молі і вогню, в якому активізація ЗПЗ здійснюється у певному вузлі на рис. 1, а далі здійснюється його поширення в напряму до приманки чи пастки, які знаходяться в центрі координат, тобто в точці О.

Згідно двох випадків можна виділити дві схеми та такі варіанти:

1) КА чи дії ЗПЗ рухаються по спіралі або по прямій, а в центрі системи координат розміщена приманка чи пастка, або об'єкт, який є метою зловмисника;

2) КА чи дії ЗПЗ рухаються по спіралі або по прямій, а в центрі системи координат розміщено приманку чи пастку, або об'єкт, який є метою зловмисника, і наявні багато об'єктів, на які спрямована одна атака;

3) КА чи дії ЗПЗ можна розглядати як об'єкт (центр освітлення) в точці О системи координат на рис. 2.10, до якого рухаються приманки по спіралі або по умовній прямій (дотична до кола);

4) КА чи дії ЗПЗ можна розглядати як об'єкт (центр освітлення) в точці О системи координат на рис. 2.10, до якого рухаються приманки по спіралі або по умовній прямій (дотична до кола) для випадку більше двох різних атак і, тоді, координатних систем буде більше двох або процес може бути відображено в паралельних площинах;

5) поєднання першого і третього варіантів;

6) поєднання другого і четвертого варіантів.

Перший і четвертий варіанти та другий і третій варіанти є не поєднувані між собою. Для різних атак можуть масштабуватись перший і другий варіанти.

Узагальнення введених паралельних площин та координатної площини, в яких зображено КА чи дії ЗПЗ та об'єкти корпоративних мереж включно з компонентами обманних систем з приманками і пастками, є поверхнями атак,

На рис. 7 рух об'єктів може здійснювати за двома траєкторіями.

Перший тип траєкторії стосується руху, який забезпечує поперечну орієнтацію і за своєю суттю є рухом по колу, тобто по дотичній до кола. Такий рух по умовній прямій характеризує те, що КА чи дії ЗПЗ не можуть досягти, не бачать об'єкту, який є метою зловмисника, або не можуть увійти в інші вузли комп'ютерних мереж),

Другий тип траєкторії стосується руху, який формує спіраль в напряму до центра системи координат, тобто до об'єкту, який цікавить зловмисника або який цікавить обманну систему з приманками і пастками, якщо розглядати такий рух в контексті двох потенційних схем для об'єктів, що зображені на рис. 1.

Таким чином, конфігурація вузлів у співвіднесенні з проникненням через засоби доступу корпоративних мереж до глобальних мереж або з середини системи в контексті використання алгоритму молі і вогню для КА і дій ЗПЗ зводиться до однієї з двох схем представлення і може бути реалізована в архітектурі обманних систем з приманками і пастками. Дві схеми представлення, в яких відмінність полягає в трактуванні центру системи координат, можуть бути узагальнені до однієї схеми або розглядатись в тривимірному просторі, де на координатній площині подано одну схему і третя координата буде надавати можливість формувати ще одну площину для другої схеми. Тоді, дві схеми будуть поєднані рух КА і дій ЗПЗ з однієї сторони буде рухом до приманок і пасток, а з другої сторони рух приманок і пасток до впливів КА і дій ЗПЗ. Підтримка зміни активності приманок і пасток в обманних системах дасть змогу забезпечити тривалу взаємодію з впливами в корпоративних мережах, які породжені КА чи діями ЗПЗ. Оптимізація при застосуванні популяційних алгоритмів в архітектурі обманних систем з приманками і пастками, зокрема алгоритму молі і вогню, для формування послідовності наступних кроків при здійсненні КА та дій ЗПЗ дає змогу уникнути повного перебору варіантів, не допустити швидкої збіжності обраних кроків при триваючих впливах та скоригувати послідовність кроків з врахуванням поточних змін в оточуючому середовищі корпоративних мереж.

ЕКСПЕРИМЕНТ

В архітектурі обманних систем реалізовано різнотипні обманні технології. Розглянемо експериментальні дослідження щодо зміни сервера.

Для дослідження алгоритму зміни сервера в розподіленій обманній системі було розгорнуто експериментальне середовище на базі програмного емулятора Cisco Packet Tracer, що відтворює типову інфраструктуру корпоративної мережі малого або середнього підприємства. Досліджувана топологія мережі складалась із трьох комутаторів другого рівня, об'єднаних у ієрархічну структуру типу core-distribution, та одного маршрутизатора для забезпечення міжмережної взаємодії між віртуальними локальними мережами.

Мережна інфраструктура була сегментована на три віртуальні локальні мережі відповідно до функціонального призначення: VLAN 10 призначено для управлінських завдань з адресним простором 192.168.10.0/24, VLAN 20 виділено для продуктивних систем з адресацією 192.168.20.0/24, а VLAN 30 зарезервовано для середовища розробки з діапазоном адрес 192.168.30.0/24. Кожна віртуальна мережа містила по три робочі станції, що забезпечувало рівномірний розподіл обчислювальних ресурсів між сегментами.

Досліджувану мережну інфраструктуру у якій здійснювалось проведення експериментів зображено на рис. 8.

ISSN 2219-9365

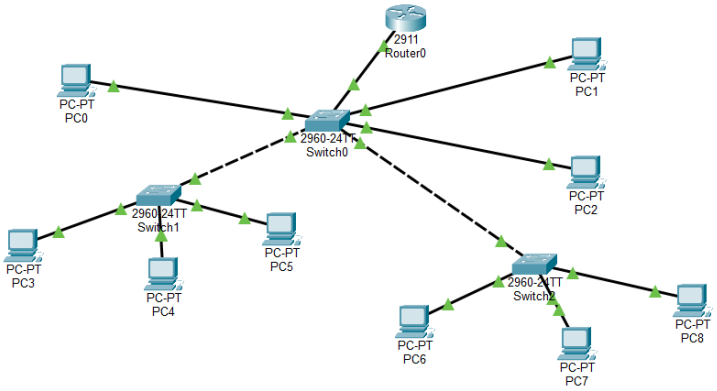

Рис. 8. Мережна інфраструктура для проведення експериментів

Початкове розміщення серверного вузла було здійснено на робочій станції PC5 з IP-адресою 192.168.20.15, що належав до сегмента VLAN 20.

Для кількісної оцінки мережних характеристик між серверним вузлом PC5 та всіма іншими робочими станціями в мережі було проведено серію систематичних вимірювань з використанням стандартних діагностичних утиліт. Метрика затримки Round Trip Time визначалась за допомогою протоколу ICMP через надсилання послідовності з десяти пакетів розміром 32 байти до кожного цільового вузла з подальшим усередненням отриманих значень часу відгуку. Кількість проміжних вузлів на шляху передачі даних встановлювалась методом трасування маршруту з реєстрацією кожного маршрутизатора або комутатора третього рівня, через які проходить пакет від джерела до призначення. З метою симуляції типових затримок корпоративної мережі на інтерфейсах міжсегментних з'єднань було застосовано обмеження пропускної здатності шляхом зміни режиму дуплексу з повного (full-duplex) на напівдуплексний (half-duplex), що призводить до збільшення часу передачі даних через необхідність арбітражу доступу до середовища. Додатково на магістральних trunk-портах між комутаторами SW1, SW2 та SW3 швидкість інтерфейсів було знижено зі стандартних 1 Гбіт/с до 100 Мбіт/с, що імітує застаріле обладнання або перевантажені канали в реальних умовах експлуатації.

Результати вимірювань продемонстрували очікувану залежність між топологічною віддаленістю вузлів та значеннями мережних метрик. Для вузлів у межах одного сегмента VLAN 20 середній час відгуку становив від 0.4 до 0.5 мілісекунди при кількості хопів від одного до двох, що відповідає прямій комутації на рівні L2. Міжсегментні з'єднання до вузлів VLAN 10 та VLAN 30 характеризувались RTT у діапазоні 2.8-3.7 мілісекунди та чотирма хопами, що обумовлено необхідністю проходження через центральний комутатор, маршрутизатор та розподільний комутатор цільового сегмента.

Для формалізації задачі вибору оптимального цільового вузла зміни сервера у розподіленій системі було розроблено комбіновану метрику відстані, що враховує як часові характеристики мережевого з'єднання, так і топологічну складність маршруту. Метрика визначалась за формулою:

$$D = \alpha \cdot \left(\frac{RTT}{RTT_{max}}\right) + \beta \cdot \left(\frac{Hops}{Hops_{max}}\right), \tag{30}$$

де D представляє нормалізовану відстань від поточного серверного вузла до потенційного кандидата на зміну, RTT позначає виміряний середній час кругового обігу пакета в мілісекундах, $Hops$ є кількістю транзитних вузлів на маршруті, RTT_{max} та $Hops_{max}$ виступають максимальними значеннями відповідних параметрів у межах досліджуваної мережі для забезпечення нормалізації, а коефіцієнти α та β визначають вагові внески кожної компоненти у підсумкову метрику з умовою $\alpha + \beta = 1$.

У контексті даного дослідження було встановлено обрано $\alpha = 0{,}7$ та $\beta = 0{,}3$, що надає пріоритет часовим характеристикам передачі даних над топологічною відстанню, оскільки для більшості розподілених застосунків саме латентність виступає критичним фактором продуктивності.

Було досліджено два альтернативних варіанти вибору цільового вузла для зміни серверного компонента. Перший варіант передбачав пошук глобального оптимуму шляхом мінімізації метрики D серед усіх доступних вузлів мережі, незалежно від їх сегментної приналежності. За результатами обчислень найменше значення метрики D = 0.157 було зафіксовано для вузла PC4 з IP-адресою 192.168.20.14, який знаходиться в тому самому сегменті VLAN 20, що й початковий сервер. Така конфігурація забезпечувала RTT на рівні 0.8 мілісекунди та потребувала лише двох хопів для досягнення, що робить цей варіант оптимальним з точки зору мінімізації мережних додаткових витрат.

Другий варіант розглядав ситуацію, коли зміна мала відбутися до альтернативного сегмента мережі з міркувань балансування навантаження або диверсифікації ризиків відмови. У цьому випадку множина потенційних кандидатів обмежувалася вузлами, що не належали до VLAN 20. Застосування критерію

мінімізації до цієї обмеженої множини виявило вузол PC3 з адресою 192.168.10.13 як оптимальний вибір з метрикою D = 0.867. Цей вузол належить до управлінського сегмента VLAN 10 і характеризувався RTT у 12.20 мілісекунди та п’ятьма хопами, що є найкращим показником серед міжсегментних з'єднань.

Теплову карта мережних метрик для оцінки зміни сервера наведено на рис. 9.

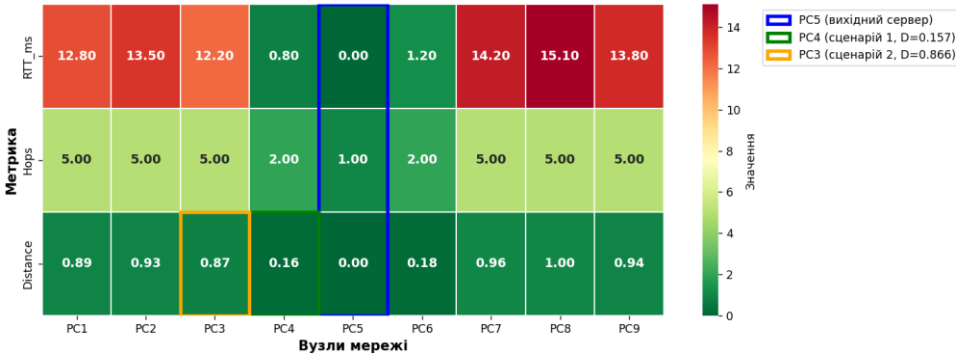

Рис. 9. Теплова карта мережевих метрик для оцінки зміни сервера

Аналіз розподілу значень метрики відстані між віртуальними мережами показав, що середня відстань до вузлів VLAN 10 становить 0.897, до VLAN 20 – 0.170 (без урахування самого PC5), а до VLAN 30 – 0.967. Така диференціація пояснюється особливостями фізичної топології та налаштувань мережевого обладнання, де вузли VLAN 30 підключені через найбільш віддалений розподільний комутатор SW3 з найвищими значеннями RTT у діапазоні 13.8-15.1 мілісекунди, що додає додаткову латентність у маршрут передачі даних та призводить до максимальних значень комбінованої метрики серед усіх сегментів мережі.

Таким чином запропонована комбінована метрика враховує як часові характеристики з’єднань, так і топологічну віддаленість вузлів, забезпечуючи адаптивність рішень через налаштування вагових коефіцієнтів відповідно до поточних потреб системи, що дозволяє визначити оптимальний варіант зміни сервера у розподілених обманних системах.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

Проблема покращення виявлення КА та дій ЗПЗ в корпоративних мережах залишається актуальною та є багатоаспектною в контексті показників, параметрів тощо, які можуть розглядатись такими для удосконалення їх безпосередньо або оновлення і які впливатимуть на вирішення проблеми. Для покращення виявлення КА та дій ЗПЗ в корпоративних мережах можуть бути оновлені або удосконалені моделі атак, архітектура обманних систем з приманками і пастками, синтез нових підсистем в архітектурі обманних систем, формування різних послідовностей наступних кроків систем із можливістю їх зміни в процесі виконання тощо.

Архітектура обманних систем з приманками і пастками повинна надавати можливість будувати засоби, які будуть адаптивними, гнучкими, самоорганізованими, прийматимуть рішення щодо своїх подальших кроків та щодо виявлення, а також організовуватимуть колективну робота агентів. Така архітектура обманних систем з приманками і пастками в поєднанні з відомостями в ній про наявні ресурси комп’ютерних мереж дасть перевагу над засобами зловмисників. Отримані числові характеристики елементів та компонентів обманних систем дають змогу здійснювати оцінювання їх поточного стану для визначення подальших кроків. Вони є частиною загальної сукупності характеристик і основною частиною для керування приманками і пастками під час здійснення атак.

В архітектурі обманних систем з приманками і пастками запропоновано синтезувати популяційні алгоритми, зокрема алгоритм молі і вогню, для здійснення оптимізації наступних кроків систем та підтримки функціонування систем тривалий час в процесі протидії атакам. Встановлено, що конфігурація вузлів мереж у співвіднесенні з проникненням через засоби доступу корпоративних мереж до глобальних мереж або з середини системи в контексті використання алгоритму молі і вогню для КА і дій ЗПЗ зводиться до однієї з двох схем представлення і може бути реалізована в архітектурі обманних систем з приманками і пастками. Дві схеми представлення, в яких відмінність полягає в трактуванні центру системи координат, можуть бути узагальнені до однієї схеми або розглядатись в тривимірному просторі, де на координатній площині подано одну схему і третя координата буде надавати можливість формувати ще одну площину для другої схеми. Підтримка зміни активності приманок і пасток в обманних системах дасть змогу забезпечити тривалу взаємодію з впливами в корпоративних мережах, які породжені КА чи діями ЗПЗ. Оптимізація при застосуванні популяційних алгоритмів в архітектурі обманних систем з приманками і пастками, зокрема алгоритму молі і вогню, для формування послідовності наступних кроків при здійсненні КА та дій ЗПЗ забезпечує уникнення повного перебору варіантів, швидкої збіжності обраних кроків при триваючих впливах

та зміну послідовності кроків з врахуванням поточних змін в оточуючому середовищі корпоративних мереж.

Напрямами подальших досліджень є адаптація популяційних алгоритмів в архітектуру обманних систем з приманками і пастками, а також розроблення методу організації функціонування таких систем згідно запропонованої концептуальної архітектури.

Література

1. Kashtalian, A., Ścisło, Ł., Rucki, R., Lysenko, S., Sachenko, A., Savenko, B., Savenko, O., & Nicheporuk, A. (2025). Control and Decision-Making in Deceptive Multi-Computer Systems Based on Previous Experience for Cybersecurity of Critical Infrastructure. *Applied Sciences*, *15*(22), 12286. https://doi.org/10.3390/app152212286

2. Dawood, M., Tu, S., Xiao, C., Alasmary, H., Waqas, M., & Rehman, S. U. (2023). Cyberattacks and Security of Cloud Computing: A Complete Guideline. *Symmetry*, *15*(11), 1981. https://doi.org/10.3390/sym15111981

3. Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. *Electronics*, *12*(6), 1333. https://doi.org/10.3390/electronics12061333

4. Shah, S. S. H., Ahmad, A. R., Jamil, N., & Khan, A. u. R. (2022). Memory Forensics-Based Malware Detection Using Computer Vision and Machine Learning. *Electronics*, *11*(16), 2579. https://doi.org/10.3390/electronics11162579

5. Malik, E. F., Khaw, K. W., Belaton, B., Wong, W. P., & Chew, X. (2022). Credit Card Fraud Detection Using a New Hybrid Machine Learning Architecture. *Mathematics*, *10*(9), 1480. https://doi.org/10.3390/math10091480

6. Gadal, S., Mokhtar, R., Abdelhaq, M., Alsaqour, R., Ali, E. S., & Saeed, R. (2022). Machine Learning-Based Anomaly Detection Using K-Mean Array and Sequential Minimal Optimization. *Electronics*, *11*(14), 2158. https://doi.org/10.3390/electronics11142158

7. Akhtar, M. S., & Feng, T. (2022). Malware Analysis and Detection Using Machine Learning Algorithms. *Symmetry*, *14*(11), 2304. https://doi.org/10.3390/sym14112304

8. Alashhab, A. A., Zahid, M. S. M., Azim, M. A., Daha, M. Y., Isyaku, B., & Ali, S. (2022). A Survey of Low Rate DDoS Detection Techniques Based on Machine Learning in Software-Defined Networks. *Symmetry*, *14*(8), 1563. https://doi.org/10.3390/sym14081563

9. Saghezchi, F. B., Mantas, G., Violas, M. A., de Oliveira Duarte, A. M., & Rodriguez, J. (2022). Machine Learning for DDoS Attack Detection in Industry 4.0 CPPSs. *Electronics*, *11*(4), 602. https://doi.org/10.3390/electronics11040602

10. Najafi Mohsenabad, H., & Tut, M. A. (2024). Optimizing Cybersecurity Attack Detection in Computer Networks: A Comparative Analysis of Bio-Inspired Optimization Algorithms Using the CSE-CIC-IDS 2018 Dataset. *Applied Sciences*, *14*(3), 1044. https://doi.org/10.3390/app14031044

11. Ahmad, I., Ul Haq, Q. E., Imran, M., Alassafi, M. O., & AlGhamdi, R. A. (2022). An Efficient Network Intrusion Detection and Classification System. *Mathematics*, *10*(3), 530. https://doi.org/10.3390/math10030530

12. Alnajim, A. M., Habib, S., Islam, M., Thwin, S. M., & Alotaibi, F. (2023). A Comprehensive Survey of Cybersecurity Threats, Attacks, and Effective Countermeasures in Industrial Internet of Things. *Technologies*, *11*(6), 161. https://doi.org/10.3390/technologies11060161

13. Safitra, M. F., Lubis, M., & Fakhrurroja, H. (2023). Counterattacking Cyber Threats: A Framework for the Future of Cybersecurity. *Sustainability*, *15*(18), 13369. https://doi.org/10.3390/su151813369

14. Aslam, M. M., Tufail, A., Kim, K.-H., Apong, R. A. A. H. M., & Raza, M. T. (2023). A Comprehensive Study on Cyber Attacks in Communication Networks in Water Purification and Distribution Plants: Challenges, Vulnerabilities, and Future Prospects. *Sensors*, *23*(18), 7999. https://doi.org/10.3390/s23187999

15. Genuario, F., Santoro, G., Giliberti, M., Bello, S., Zazzera, E., & Impedovo, D. (2024). Machine Learning-Based Methodologies for Cyber-Attacks and Network Traffic Monitoring: A Review and Insights. *Information*, *15*(11), 741. https://doi.org/10.3390/info15110741

16. Ding, S., Liu, G., Yin, L., Wang, J., & Li, Z. (2024). Detection of Cyber-Attacks in a Discrete Event System Based on Deep Learning. *Mathematics*, *12*(17), 2635. https://doi.org/10.3390/math12172635

17. Lallie, H. S., Thompson, A., Titis, E., & Stephens, P. (2025). Analysing Cyber Attacks and Cyber Security Vulnerabilities in the University Sector. *Computers*, *14*(2), 49. https://doi.org/10.3390/computers14020049

References

1. Kashtalian, A., Ścisło, Ł., Rucki, R., Lysenko, S., Sachenko, A., Savenko, B., Savenko, O., & Nicheporuk, A. (2025). Control and Decision-Making in Deceptive Multi-Computer Systems Based on Previous Experience for Cybersecurity of Critical Infrastructure. *Applied Sciences*, *15*(22), 12286. https://doi.org/10.3390/app152212286

2. Dawood, M., Tu, S., Xiao, C., Alasmary, H., Waqas, M., & Rehman, S. U. (2023). Cyberattacks and Security of Cloud Computing: A Complete Guideline. *Symmetry*, *15*(11), 1981. https://doi.org/10.3390/sym15111981

3. Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. *Electronics*, *12*(6), 1333. https://doi.org/10.3390/electronics12061333

4. Shah, S. S. H., Ahmad, A. R., Jamil, N., & Khan, A. u. R. (2022). Memory Forensics-Based Malware Detection Using Computer Vision and Machine Learning. *Electronics*, *11*(16), 2579. https://doi.org/10.3390/electronics11162579

5. Malik, E. F., Khaw, K. W., Belaton, B., Wong, W. P., & Chew, X. (2022). Credit Card Fraud Detection Using a New Hybrid Machine Learning Architecture. *Mathematics*, *10*(9), 1480. https://doi.org/10.3390/math10091480

6. Gadal, S., Mokhtar, R., Abdelhaq, M., Alsaqour, R., Ali, E. S., & Saeed, R. (2022). Machine Learning-Based Anomaly Detection Using K-Mean Array and Sequential Minimal Optimization. *Electronics*, *11*(14), 2158. https://doi.org/10.3390/electronics11142158

7. Akhtar, M. S., & Feng, T. (2022). Malware Analysis and Detection Using Machine Learning Algorithms. *Symmetry*, *14*(11), 2304. https://doi.org/10.3390/sym14112304

8. Alashhab, A. A., Zahid, M. S. M., Azim, M. A., Daha, M. Y., Isyaku, B., & Ali, S. (2022). A Survey of Low Rate DDoS Detection Techniques Based on Machine Learning in Software-Defined Networks. *Symmetry*, *14*(8), 1563. https://doi.org/10.3390/sym14081563

9. Saghezchi, F. B., Mantas, G., Violas, M. A., de Oliveira Duarte, A. M., & Rodriguez, J. (2022). Machine Learning for DDoS Attack Detection in Industry 4.0 CPPSs. *Electronics*, *11*(4), 602. https://doi.org/10.3390/electronics11040602

10. Najafi Mohsenabad, H., & Tut, M. A. (2024). Optimizing Cybersecurity Attack Detection in Computer Networks: A Comparative Analysis of Bio-Inspired Optimization Algorithms Using the CSE-CIC-IDS 2018 Dataset. *Applied Sciences*, *14*(3), 1044. https://doi.org/10.3390/app14031044

11. Ahmad, I., Ul Haq, Q. E., Imran, M., Alassafi, M. O., & AlGhamdi, R. A. (2022). An Efficient Network Intrusion Detection and Classification System. *Mathematics*, *10*(3), 530. https://doi.org/10.3390/math10030530

12. Alnajim, A. M., Habib, S., Islam, M., Thwin, S. M., & Alotaibi, F. (2023). A Comprehensive Survey of Cybersecurity Threats, Attacks, and Effective Countermeasures in Industrial Internet of Things. *Technologies*, *11*(6), 161. https://doi.org/10.3390/technologies11060161

13. Safitra, M. F., Lubis, M., & Fakhrurroja, H. (2023). Counterattacking Cyber Threats: A Framework for the Future of Cybersecurity. *Sustainability*, *15*(18), 13369. https://doi.org/10.3390/su151813369

14. Aslam, M. M., Tufail, A., Kim, K.-H., Apong, R. A. A. H. M., & Raza, M. T. (2023). A Comprehensive Study on Cyber Attacks in Communication Networks in Water Purification and Distribution Plants: Challenges, Vulnerabilities, and Future Prospects. *Sensors*, *23*(18), 7999. https://doi.org/10.3390/s23187999

15. Genuario, F., Santoro, G., Giliberti, M., Bello, S., Zazzera, E., & Impedovo, D. (2024). Machine Learning-Based Methodologies for Cyber-Attacks and Network Traffic Monitoring: A Review and Insights. *Information*, *15*(11), 741. https://doi.org/10.3390/info15110741

16. Ding, S., Liu, G., Yin, L., Wang, J., & Li, Z. (2024). Detection of Cyber-Attacks in a Discrete Event System Based on Deep Learning. *Mathematics*, *12*(17), 2635. https://doi.org/10.3390/math12172635

17. Lallie, H. S., Thompson, A., Titis, E., & Stephens, P. (2025). Analysing Cyber Attacks and Cyber Security Vulnerabilities in the University Sector. *Computers*, *14*(2), 49. https://doi.org/10.3390/computers14020049