УДК 004.681
DOI https://doi.org/10.26661/2786-6254-2024-1-06

МЕТОДОЛОГІЯ РАННЬОГО ВИЯВЛЕННЯ ПОТЕНЦІЙНИХ КАНАЛІВ УРАЖЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

Борисенко Б. В.
аспірант кафедри інформаційних управляючих систем і технологій
Ужгородський національний університет
вул. Університетська, 14, Ужгород, Україна
orcid.org/0000-0003-1320-5083

Ключові слова: *комп'ютерна система, небезпека, ураження, канал, методологія, вторгнення, атака.*

У статті проведено розробку методології раннього виявлення потенційних каналів ураження комп'ютерних систем. Зазначається, що компоненти багатоагентної системи виявлення потенційних каналів ураження комп'ютерних систем – це взаємодіючі між собою агенти, які спільно вирішують загальне завдання виявлення вторгнень у комп'ютерну систему. Архітектура зазначеної системи включає один або кілька екземплярів агентів різних типів, спеціалізованих на вирішення задачі виявлення потенційних каналів ураження комп'ютерних систем. Агенти розподілені по елементам системи, що захищається, спеціалізовані за типами розв'язуваних завдань і взаємодіють один із одним з метою обміну інформацією та прийняття узгоджених рішень. Наголошується, що у прийнятій архітектурі у явному вигляді відсутній «центр управління» сімейством агентів – залежно від ситуації провідним може ставати будь-який із агентів, котрий ініціює і (або) реалізує функції кооперації й управління. У разі потреби агенти можуть як клонуватися (утворювати нові сутності), так і припиняти своє функціонування. Залежно від ситуації (виду та кількості атак на комп'ютерну систему, наявності обчислювальних ресурсів для виконання функцій захисту) може знадобитися генерація кількох екземплярів агентів кожного класу. Розроблена методологія призначена для підвищення ефективності ідентифікації потенційних каналів ураження комп'ютерних систем, що дозволить забезпечити необхідний рівень цілісності та доступності інформації. Методологія раннього виявлення потенційних каналів ураження комп'ютерних систем базується на концепції мультикомп'ютерних систем із контролером прийняття рішень для виявлення потенційних каналів ураження та протидії шкідливим програмам і комп'ютерним атакам. Застосування розробленої методології раннього виявлення потенційних каналів ураження комп'ютерних систем дозволяє підвищити гнучкість системи та забезпечити необхідний рівень захищеності інформаційних систем.

METHODOLOGY FOR EARLY DETECTION OF POTENTIAL CHANNELS OF DAMAGE TO COMPUTER SYSTEMS

Borysenko B. V.
Postgraduate Student at the Department of Information Management Systems and Technologies
Uzhhorod National University
Universytetska str., 14, Uzhhorod, Ukraine
orcid.org/0000-0003-1320-5083

Key words: *computer system, danger, defeat, channel, methodology, intrusion, attack.*

Within the scope of the article, the methodology for early detection of potential channels of damage to computer systems has been developed. It is noted that the components of the multi-agent system for detecting potential channels of damage to computer systems are interacting agents that jointly solve the common task of detecting intrusions into the computer system. The architecture of the specified system includes one or more instances of agents of various types, specialized in solving the problem of identifying potential channels of damage to computer systems. Agents are distributed among the elements of the protected system, specialized in the types of tasks to be solved and interact with each other in order to exchange information and make agreed decisions. It is emphasized that the adopted architecture clearly lacks a "control center" for the family of agents – depending on the situation, any of the agents who initiate and (or) implement cooperation and management functions can become the leader. If necessary, agents can both clone (form new entities) and stop their functioning. Depending on the situation (the type and number of attacks on the computer system, the availability of computing resources to perform protection functions), it may be necessary to generate several instances of agents of each class. The developed methodology is intended to increase the efficiency of identification of potential channels of damage to computer systems, which will ensure the necessary level of integrity and availability of information. The methodology for early detection of potential damage channels of computer systems is based on the concept of multi-computer systems with a decision-making controller to identify potential damage channels and counter malicious programs and computer attacks. Application of the developed methodology for early detection of potential channels of damage to computer systems allows to increase the flexibility of the system and ensure the necessary level of security of information systems.

Вступ. Користувачам комп'ютерних систем потрібні системи раннього виявлення потенційних каналів ураження, шкідливого програмного забезпечення та комп'ютерних атак, котрі дозволять, крім забезпечення безпеки, визначити імовірність вторгнення на різних етапах. Серед систем виявлення потенційних каналів ураження, шкідливих програм і комп'ютерних атак є системи, які, окрім виявлення загроз, створюють хибні цілі для атак у комп'ютерних системах, що дозволяє адміністраторам таких систем відстежувати процеси, які є зловмисними або виходять за рамками встановлених функцій [1]. Таким чином, системи, орієнтовані на виявлення потенційних каналів ураження, що пройшли певні ступені захисту та використовують традиційні засоби та системи запобігання, виявлення та протидії вторгнень, призначення яких і можливі варіанти конфігурації використання яких відомі зловмисникам, є перспективним напрямком розвитку. Серед таких систем особливе місце у класифікації посідають системи запобігання, виявлення та протидії з певним набором приманок і пасток для шкідливих програм і комп'ютерних атак. Їх використання створює для зловмисника помилкові цілі для атак і дозволяє зберігати інформацію про такі атаки та поширення шкідливих програм на комп'ютерних станціях у мережі.

Щоб підвищити ефективність систем виявлення та протидії шкідливим програмам і комп'ютерним атакам за допомогою приманок і пасток, необхідно інтегрувати ці засоби у складні системи, що включають всі комп'ютерні станції у мережі, й організувати їх роботу таким чином,

щоб вони могли спільно і без втручання користувача реагувати на шкідливі й аномальні процеси. Таким чином, необхідно побудувати не просто одну приманку і пастку на конкретній комп'ютерній станції, а мережу приманок і пасток, щоб забезпечити комплексний захист комп'ютерної системи на етапі, коли комп'ютерні атаки зуміли пройти через брандмауер і шкідливе програмне забезпечення зуміло подолати сканування антивірусними засобами та системами. Така система приманок і пасток може бути комбінованою, і для досягнення ефективного результату вона повинна містити тіньові приманки та пастки, які дозволять встановити і відстежити поведінку зловмисника під час атаки, а також виявляти шкідливі програми та комп'ютерні атаки з більшою ймовірністю. Ефективність таких інструментів залежить від організаційної складової частини системи.

Огляд літератури. Формулювання наукової думки щодо безпеки комп'ютерних систем є різнорідним і масштабним. У сучасній науковій площині з'являються роботи, присвячені дослідженням каналів ураження та попередження злому для підвищення рівня кібербезпеки.

С.Ф. Гончар та М.Ю. Комаров [2] розробили таксономію кіберзагроз інформаційно-телекомунікаційним мережам об'єктів критичної інфраструктури. Також авторами складено матрицю залежності інформаційних об'єктів захисту від типу потенційних загроз, що можуть на них впливати, та схильності до конкретних загроз, розроблено модель бази даних кіберзагроз інформаційним об'єктам захисту інформаційно-телекомунікаційних систем об'єктів критичної інфраструктури.

У [3] сформовано теоретичні положення, науково-методичні підходи та практичні рекомендації щодо формування інформаційної безпеки національної економіки.

Б.В. Петрик, В.Р. Дейнега та Г.В. Неласа [4] розглянули питання аналізу мережевого трафіку для виявлення потенційних проблем за допомогою вейвлет-аналізу з використанням вейвлету Хаара й алгоритму Малла. Також було описано процес усунення шумів у сигналі при вейвлет-аналізі.

Авторами [5] представлено метод виявлення кібератак соціальної інженерії. Підхід виявлення кібератак використовував чотири алгоритми машинного навчання (decision tree, random forest, K-nearest neighbor, extreme gradient boosting). Аналіз зосереджено на даних, зібраних із хостів мережі, які можуть служити індикаторами потенційної атаки соціальної інженерії. Емпіричні результати продемонстрували високу точність визначення.

Із зарубіжних авторів варто відзначити роботи таких науковців, як Роутер Шіджу, Шиваджі Сатьялакшмі [6], Саліх Азар, Абдулраззак Майван [7], Шафей Хамідреза, Лі Лі, Агілера Рікардо [8], Спенс Аарон, Бангай Шон [9], Лю Сяо-Хуей, Донг Цзюн, Ян Гуан-Хонг [10], Такахаші Макото, Мацумото Кохей [11], Раутер Шію, Шиваджі Сатьялакшмі [12], С. Раджан, Р. Картика [13], Тахун Алі, Арафа Мохаммад [14] та ін.

Однак, незважаючи на масштабність наукових досліджень, актуальність роботи не викликає сумнівів.

Методи. Для використання технології раннього виявлення потенційних каналів ураження комп'ютерних систем були розроблені різні види приманок і пасток, які імітують роботу реальних систем. Ринок пропонує декілька рішень, заснованих на використанні технологій обману та шкідливих програм.

Розглядається можливий варіант методології раннього виявлення потенційних каналів ураження комп'ютерних систем, зокрема архітектура системи із частковою централізацією. Для синтезу таких систем розроблено новий метод синтезу частково централізованих систем для виявлення шкідливих програм у комп'ютерному середовищі на основі аналітичних виразів, які визначають стан безпеки таких систем і їх компонентів. Методи синтезу частково централізованих систем застосовуються для виявлення потенційних каналів ураження комп'ютерних систем.

Результати. Методологія раннього виявлення потенційних каналів ураження комп'ютерних систем ґрунтується на створенні частково централізованих компонентів і систем виявлення шкідливих програм у комп'ютерних мережах. На рис. 1 наведено схему реалізації методологічної складової запропонованого підходу.

Деталізуємо кожен із наведених принципів функціонування та характеристик. Все це повинно бути синтезовано у таких системах повністю. Завдяки такому синтезу система стане самоорганізованою, адаптивною та частково централізованою.

Формування системи S із компонентів може здійснюватися на початку її інсталяції та введення в дію, за необхідності у процесі експлуатації та після включення комп'ютерних станцій у мережу. Крім того, до системи можна додавати нові компоненти або видаляти наявні. Крім того, можуть бути деякі комп'ютерні станції, на яких встановлені компоненти вимкнено на тривалий час; отже, система міститиме меншу кількість компонентів.

Комп'ютерні станції із системними компонентами можуть бути включені одночасно або у різний час. Комп'ютерні станції можна не вимикати, тобто вони можуть бути включені постійно. Ці випадки впливатимуть на формування системи S. Встановимо їх у системі, щоб її центр прийняття рішень міг розглядати ці випадки та їх варіації у

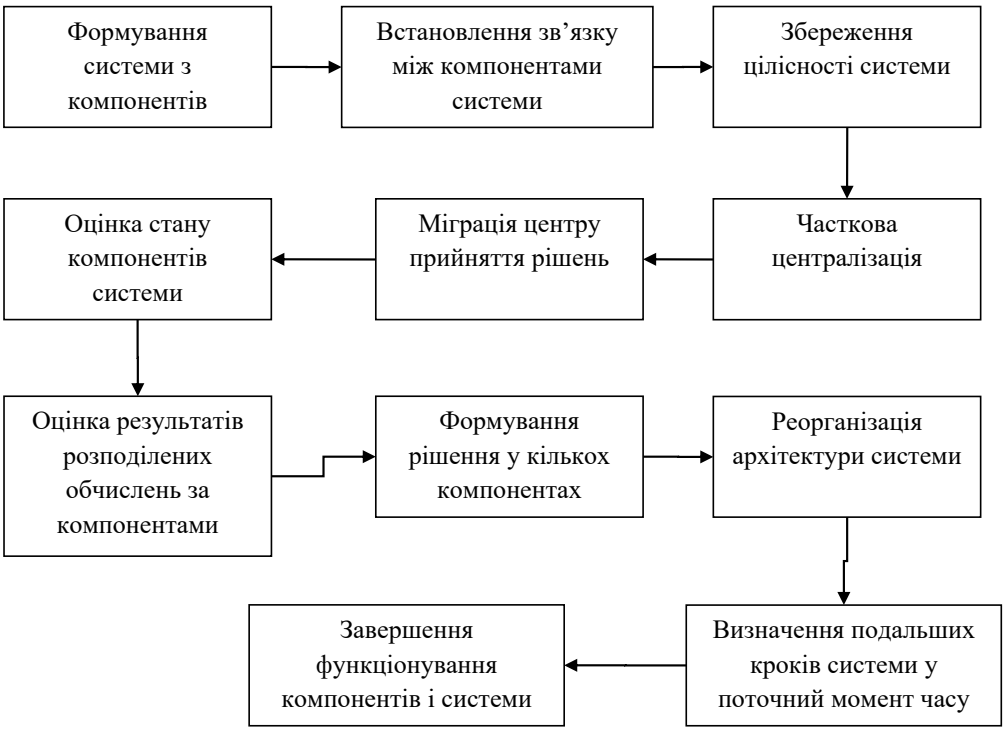

Рис. 1. Методологія раннього виявлення потенційних каналів ураження комп'ютерних систем

процесі формування та функціонування системи, а також як активну останню подію. Варіанти формування системи визначаються як сукупність,

$$M_s^{var,1} = \left\{ m_{s,1}^{var,1}, m_{s,2}^{var,1}, \ldots, m_{s,n_{M_s^{var,1}}}^{var,1} \right\}$$

де $n_{M_s^{var,1}}$ – це кількість варіантів. Наприклад, такі елементи як:

$m_{s,1}^{var,1}$ – характеризує зародження системи на її початку й активацію;

$m_{s,2}^{var,1}$ – характеризує формування системи у процесі функціонування за потреби;

$m_{s,3}^{var,1}$ – характеризує формування системи після включення комп'ютерних станцій у мережу. Опції, які встановлює множина $M_s^{var,1}$, на поточний момент часу може бути лише одна. Тобто система S аналізуватиме останній варіант свого формування. Для визначення останнього варіанта побудови системи введемо предикат на елементах множини у вигляді: $M_s^{var,1}$.Варіант, заданий множиною, на поточний момент часу може бути лише один. Тобто система аналізує останній варіант його формування.

$$P_s^{var,1}\left(m_{s,q}^{var,1}\right) = \begin{cases} 0, m_{s,q}^{var,1} - неактуальна версія \\ q, m_{s,q}^{var,1} - актуальна версія \end{cases}$$

$$q = 1,2,\ldots,n_{M_s^{var,1}}$$

Подібним чином вводимо набір варіацій за допомогою

$$M_s^{var,2} = \left\{ m_{s,1}^{var,2}, m_{s,2}^{var,2}, \ldots, m_{s,n_{M_s^{var,2}}}^{var,2} \right\}$$

де $n_{M_s^{var,2}}$ це кількість варіацій. Наприклад, такі елементи як:

$m_{s,1}^{var,2}$ – доповнення системи з новими компонентами;

$m_{s,2}^{var,2}$ – видалення компонентів із системи.

Для варіацій, заданих набором випадків $M_s^{var,2}$, вводимо предикат, значення якого буде відображати їх наявність або відсутність

$$P_s^{var,2}\left(m_{s,q}^{var,2}\right) = \begin{cases} 0, m_{s,q}^{var,2} - неактуальна версія \\ q, m_{s,q}^{var,2} - актуальна версія \end{cases}$$

$$q = 1,2,\ldots,n_{M_s^{var,2}}$$

Подібним чином вводимо набір варіацій за допомогою

$$M_s^{var,3} = \left\{ m_{s,1}^{var,3}, m_{s,2}^{var,3}, \ldots, m_{s,n_{M_s^{var,3}}}^{var,3} \right\}$$

де $n_{M_s^{var,3}}$ – це кількість варіацій. Наприклад, такі елементи:

$m_{s,1}^{var,3}$ –комп'ютерні станції, у яких є компоненти системи, включені у певний час;

$m_{s,2}^{var,3}$ – комп'ютерні станції, у яких є системні компоненти, які вмикаються у різний час;

$m_{s,3}^{var,3}$ – комп'ютерні станції, у яких компоненти системи не вимикаються протягом усього часу роботи системи.

Для варіацій, заданих набором випадків $M_s^{var,3}$, вводимо предикат, значення якого буде відображати їх наявність або відсутність:

$$P_s^{var,3}\left(m_{s,q}^{var,3}\right)=\begin{cases}0, m_{s,q}^{var,3}-\textit{неактуальна версія}\\ q, m_{s,q}^{var,3}-\textit{актуальна версія}\end{cases}$$

$$q=1,2,\ldots,n_{M_s^{var,3}}$$

Подібним чином вводимо набір варіацій за допомогою

$$M_s^{var,4}=\left\{m_{s,1}^{var,4}, m_{s,2}^{var,4},\ldots, m_{s,n_{M_s^{var,4}}}^{var,4}\right\}$$

де $n_{M_s^{var,4}}$ – це кількість варіацій. Наприклад, такі елементи:

$m_{s,1}^{var,4}$ – частина комп'ютерних станцій, у яких встановлені компоненти, може бути вимкнена на тривалий час, система буде містити меншу частину компонентів і водночас може відбуватися її поточне формування, викликане певними подіями без цих компонентів;

$m_{s,2}^{var,4}$ – модифікація системи не відбувалася без компонентів, які знаходилися у вимкнених комп'ютерних станціях.

Для варіації, заданої набором випадків $M_s^{var,4}$, введемо предикат, значення якого буде відображати їх наявність або відсутність:

$$P_s^{var,4}\left(m_{s,q}^{var,4}\right)=\begin{cases}0, m_{s,q}^{var,4}-\textit{неактуальна версія}\\ q, m_{s,q}^{var,4}-\textit{актуальна версія}\end{cases}$$

$$q=1,2,\ldots,n_{M_s^{var,4}}$$

Наведені формули описують стадію формування системи S та конкретизують її варіанти. Результати обчислення предикату є частиною вхідних даних для центру прийняття рішень системи. Після встановлення всіх компонентів системи на комп'ютерних станціях у мережі, враховуючи компоненти із центром прийняття рішень і без нього, при першому запуску системи компоненти із центром прийняття рішень перевіряють значення предикатів для різних елементів набору $M_s^{var,1}$ і встановлюють, що всі величини дорівнюють нулю. Далі система самостійно, без користувача чи адміністратора, почне первинне формування своїх компонентів із наявних підмножин функцій, а після завершення такого формування перейде до поділу компонентів із центром прийняття рішень на активні та неактивні.

Для забезпечення зв'язку між компонентами у системі S організуємо зв'язок між компонентами не лише за допомогою стандартної розсилки повідомлень із відповідною кількістю повідомлень підтвердження, а й із послідовним додаванням до них певних завдань, результат яких відомий у компонентів, які планують відправити основне повідомлення або завдання, а також провести аналіз часу, витраченого між відправленням першого запиту на підключення до отримання результатів тестового завдання. Загалом, уся система S діятиме як один великий датчик, котрий реагуватиме на зміни у роботі її частин, включаючи зв'язок між компонентами. Якщо всі компоненти вимкнути одночасно, то вони фіксують завдання, виконання якого вони повинні виконати після наступного включення. Однак може статися так, що комп'ютерна станція аварійно вимкнеться, і така фіксація певного контрольного завдання не відбудеться. Таким чином, впровадження резервування в організацію зв'язку між компонентами вимагає врахування варіантів з увімкненими та вимкненими комп'ютерними станціями та синхронізації часу, протягом якого компоненти активні та встановлюють зв'язок між собою. Тому введемо набір варіантів

$$M_s^{var,5}=\left\{m_{s,1}^{var,5}, m_{s,2}^{var,5},\ldots, m_{s,n_{M_s^{var,5}}}^{var,5}\right\}$$

де $n_{M_s^{var,5}}$ – кількість варіантів, які виникають, коли вводиться резервування для організації зв'язку між компонентами.

Елементи набору визначаються як:

$m_{s,1}^{var,5}$ –комп'ютерні станції, у якій системі комплектуючі наявні, включені одночасно;

$m_{s,2}^{var,5}$ – комп'ютерні станції, у яких компоненти вмикаються у різний час, частина може вимкнутися через певний час роботи, а певна частина може ввімкнутися після цього часу або не вмикатися взагалі протягом тривалого часу.

Відповідно, компоненти системи S також повинні бути активними лише тоді, коли комп'ютерні станції увімкнені та функціонують. Набір опцій визначається як:

$$M_s^{var,6}=\left\{m_{s,1}^{var,6}, m_{s,2}^{var,6},\ldots, m_{s,n_{M_s^{var,6}}}^{var,6}\right\}$$

де $n_{M_s^{var,6}}$ – кількість варіантів, що виникають при завершенні роботи комп'ютерних станцій, на яких встановлені компоненти системи.

Елементи множини визначаються таким чином:

$m_{s,1}^{var,6}$ – комп'ютерні станції, у яких системні компоненти присутні, вимкнено правильно в один і той самий час;

$m_{s,2}^{var,6}$ – комп'ютерні станції, у яких є компоненти системи, відключені аварійно в однаковий час;

$m_{s,3}^{var,6}$ – комп'ютерні станції, що працюють правильно;

$m_{s,4}^{var,6}$ –комп'ютерні станції, в яких є компоненти системи, відключені у різний час, частково правильно, частково аварійно.

За наведеними множинами можна сформувати двоелементні множини, які характеризують події зв'язку у системі залежно від комп'ютерних станцій таким чином:

$$\left\{m_{s,1}^{var,5};m_{s,1}^{var,6}\right\};\left\{m_{s,1}^{var,5};m_{s,2}^{var,6}\right\};\left\{m_{s,1}^{var,5};m_{s,3}^{var,6}\right\}$$

$$\left\{m_{s,1}^{var,5};m_{s,4}^{var,6}\right\};\left\{m_{s,2}^{var,5};m_{s,1}^{var,6}\right\};\left\{m_{s,2}^{var,5};m_{s,2}^{var,6}\right\}$$

$$\left\{m_{s,2}^{var,5};m_{s,3}^{var,6}\right\};\left\{m_{s,2}^{var,5};m_{s,4}^{var,6}\right\}$$

Для окремих комп'ютерних станцій необхідно розробляти подібні задачі комплексно, оскільки відповідно до них буде забезпечений зв'язок між окремими компонентами у системі. Загалом у системі зв'язок між компонентами та надсиланням повідомлень буде встановлюватися відповідно до таких відносин: «один до всіх» ($m_{s,1}^{var,7}$); «всі до одного» ($m_{s,2}^{var,7}$); «кожному інший» ($m_{s,3}^{var,7}$); «один до певної кількості, але не до всіх» ($m_{s,4}^{var,7}$); «певне число, але не всі, до одного» ($m_{s,5}^{var,7}$); «певне число, але не всі, до певної кількості, але не всім» ($m_{s,6}^{var,7}$). Визначимо ці відношення як набір

$$M_s^{var,7}=\left\{m_{s,1}^{var,7},m_{s,2}^{var,7},\ldots,m_{s,n_{M_s^{var,7}}}^{var,7}\right\}$$

де $n_{M_s^{var,7}}$ це кількість, $n_{M_s^{var,7}}=6$.

Для визначення зв'язку між окремими комп'ютерними станціями вводимо набір параметрів

$$M_s^{var,8}=\left\{m_{s,1}^{var,8},m_{s,2}^{var,8},\ldots,m_{s,n_{M_s^{var,8}}}^{var,8}\right\}$$

де $n_{M_s^{var,8}}$ це кількість варіантів, які виникають у процесі встановлення зв'язку між комп'ютерними станціями, у яких знаходяться компоненти системи.

Елементи набору такі:

$m_{s,1}^{var,8}$ – комп'ютерна станція, у якій знаходиться включений компонент системи;

$m_{s,2}^{var,8}$ – комп'ютерна станція, у який присутній компонент системи, вимкнений правильно;

$m_{s,3}^{var,8}$ – комп'ютерна станція, у якій компонент системи, аварійно вимкнений.

За заданою множиною сформуємо двоелементні підмножини, які характеризують стан комп'ютерних станцій щодо початку та закінчення їх роботи таким чином:

$$\left\{m_{s,1}^{var,8};m_{s,2}^{var,8}\right\};\left\{m_{s,1}^{var,8};m_{s,3}^{var,8}\right\}.$$

Отже, якщо стан комп'ютерної станції, у якій присутній компонент S системи, характеризується підмножиною $\left\{m_{s,1}^{var,8};m_{s,2}^{var,8}\right\}$, тоді повідомлення, які він отримує та надсилає, будуть вважатися центром прийняття рішень, виконаними правильно. В іншому випадку, тобто для підмножини $\left\{m_{s,1}^{var,8};m_{s,3}^{var,8}\right\}$, центр прийняття рішень фіксує таку подію і при наступному включенні комп'ютерної станції обробляє додаткову спеціальну процедуру встановлення зв'язку з цим компонентом для оновлення цього компонента у системі. Крім того, при виконанні стандартної комунікаційної дії між будь-якими двома компонентами системи, незалежно від типу елемента множини $M_s^{var,7}$, виконання додаткової перевірки є обов'язковим і полягає у виконанні певного завдання компоненту, який планує встановити зв'язок, і таке ж певне завдання від компонента, з яким планується з'єднання.

Таким чином, встановлення зв'язку між компонентами системи у різних вузлах мережі буде здійснюватися з урахуванням типів зв'язків, котрі дозволяють синтез часткової централізації та додаткової перевірки легітимності компонента.

Корпоративна мережа підприємства може мати кілька сегментів. Компоненти системи S можуть бути встановлені у різних частинах мережі та віддалено у домашніх комп'ютерних станціях. У корпоративній мережі комутатори можуть вийти з ладу або можуть виникнути інші причини, які спричинять поділ системи на дві або більше непов'язаних підсистем. Тобто система у процесі функціонування може розпадатися на непов'язані між собою частини. Потім кожна із частин перетворюється на скорочену систему S і продовжує працювати, якщо в кожній із частин залишаються хоча би два активні компоненти із центром прийняття рішень. Якщо одна із частин не має активних компонентів із центром прийняття рішень і є неактивною, то компоненти цієї частини блокують роботу комп'ютерних станцій і видають відповідне повідомлення адміністратору. Якщо компоненти із центром прийняття рішень неактивні в момент певної аварійної ситуації або навмисного відокремлення другої частини, яка міститиме всі активні компоненти із центром прийняття рішень системи, то їх переведення в активний стан відбудеться після чергового сеансу зв'язку та встановлюється відсутність зв'язку активних компонентів із центром прийняття рішень. Підтримання цілісності системи S під час її роботи забезпечуватиметься періодичним обміном повідомленнями між компонентами системи згідно з відношенням із множини $M_s^{var,7}$, які будуть обрані випадковим чином. Крім цих двох випадків, що характеризують забезпечення цілісності системи, існує також випадок, пов'язаний із синтезом часткової централізації у системі S. Якщо частина активних компонентів, які містять центр прийняття рішень системи, з певних причин видалено із системи, то частина, що залишилася, почне процедуру формування системи з наявних компонентів. Але якщо таких компонентів менше двох, то всі активні компоненти, в тому числі без функціоналу із центром прийняття рішень, блокуватимуть роботу комп'ютерних станцій і видадуть відповідне повідомлення системному адміністратору. Отже, така організація забезпечення цілісності системи враховує можливість синтезу в системі S часткової централізації й адаптивності.

Система є частково централізованою, оскільки всі її компоненти поділяються на дві підмножини: підмножину компонентів, які можуть бути центром системи, і підмножину компонентів, котрі не мають функцій для забезпечення функціонування центру прийняття рішень системи. Управління всією системою відбувається від компонентів, у яких знаходиться центр прийняття рішень системи. Тому вона централізована. Часткова централізація забезпечується тим, що компоненти системи S, у яких знаходиться центр прийняття рішень системи S для прийняття рішень, розробляють пропозиції окремо в кожному з цих компонентів, тобто децентралізовано, і погоджують їх спільно. Таким чином, система не є повністю централізованою.

Більшість встановлених компонентів системи S у комп'ютерних станціях повинні містити функціональність, що забезпечує функціонування центру прийняття рішень системи. Після завершення встановлення системи здійснюється перший запуск системи з увімкненими всіма комп'ютерними станціями, на яких встановлені компоненти системи. На цьому етапі функціонування системи всі компоненти, які можуть мати центр прийняття рішень системи, братимуть участь у підготовці першого остаточного рішення щодо визначення першого кроку системи. Це рішення дозволить зменшити кількість активних компонентів центру прийняття рішень, перевівши частину з них у неактивний стан, що підвищить імовірність раннього виявлення потенційних каналів ураження системи.

Висновки. Методологія раннього виявлення потенційних каналів ураження комп'ютерних систем базується на концепції мультикомп'ютерних систем із контролером прийняття рішень для виявлення потенційних каналів ураження та протидії шкідливим програмам і комп'ютерним атакам. Для деталізації архітектури мультикомп'ютерної системи з контролером прийняття рішень щодо виявлення та протидії шкідливим програмним і комп'ютерним атакам, що відповідає запропонованому принципу синтезу таких систем, необхідно розробити концептуальну модель її архітектури. Реалізація регулятора прийняття рішень через розробку методу синтезу систем із контролером буде напрямом подальших досліджень.

ЛІТЕРАТУРА

1. Бурячок В.Л., Киричок Р.В., Складанний П.М. Основи інформаційної та кібернетичної безпеки. Київ, 2018. 320 с.
2. Гончар С.Ф., Комаров М.Ю. Безпека інформації в комп'ютерних системах та мережах об'єктів критичної інфраструктури : монографія. Київ : «Три К», 2021. 119 с.
3. Білько С.С. Формування інформаційної безпеки національної економіки : дис. … докт. філософії : 051. Національний університет «Полтавська політехніка імені Юрія Кондратюка». Полтава, 2023. 212 с.
4. Застосування вейвлет-аналізу для виявлення аномалій мережевого трафіку / Б.В. Петрик, В.Р. Дейнега, Г.В. Неласа. *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій* : Тези доповідей X Міжнародної науково-практичної конференції, 07–09 жовтня 2020 р., м. Запоріжжя/ відпов. ред. С.В. Морщавка. Запоріжжя : НУ «Запорізька політехніка», 2020. С. 163–167.
5. Бохонько О.О., Лисенко С.М. Метод виявлення кібератак на основі соціальної інженерії. З*бірник наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023».* Хмельницький, 2023. С. 38–41.
6. Rawther Shiju, Sivaji Sathyalakshmi. Entropy of a Computer Network Under Propagation of Cyber-Attacks. *International Journal of Engineering Trends and Technology.* 2023. № 71. P. 295–303. DOI: 10.14445/22315381/IJETT-V71I8P226.
7. Salih Azar, Abdulrazzaq Maiwan. Cyber security: performance analysis and challenges for cyber attacks detection. *Indonesian Journal of Electrical Engineering and Computer Science.* 2023. № 31. P. 17–63. DOI: 10.11591/ijeecs.v31.i3.pp1763-1775.
8. Shafei Hamidreza, Li Li, Aguilera Ricardo. A Comprehensive Review on Cyber-Attack Detection and Control of Microgrid Systems. 2023. DOI: 10.1007/978-3-031-20360-2_1.
9. Spence Aaron, Bangay Shaun. Security beyond cybersecurity: side-channel attacks against non-cyber systems and their countermeasures. *International Journal of Information Security*. 2022. № 21. DOI: 10.1007/s10207-021-00563-6.
10. Liu Xiao-Hui, Dong Jiuxiang, Yang Guang-Hong. Optimal DoS Attack Scheduling for Cyber-Physical Systems With Channel Hopping Scheme. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2023. P. 1–8. DOI: 10.1109/TSMC.2023.3305287.

11. Takahashi Makoto, Matsumoto Kohei. Experimental Study On The Cyber Attack Early Recognition System. *The Proceedings of the International Conference on Nuclear Engineering (ICONE).* 2023. № 30. P. 18–30. DOI:10.1299/jsmeicone.2023.30.1830.
12. Rawther Shiju, Sivaji Sathyalakshmi. Analysing the Spread of Cyber-Attacks in Computer Networks: A Simulation Study. *International Journal of Engineering Trends and Technology.* 2023. № 71. P. 26–38. DOI: 10.14445/22315381/IJETT-V71I8P203.
13. Rajan S., Karthika R. A Survey of Computational Intelligence Techniques Used for Cyber-Attack Detection. 2022. DOI: 10.1007/978-981-19-3571-8_49.
14. Tahoun Ali, Arafa Mohammad. Secure control design for nonlinear cyber–physical systems under DoS, replay, and deception cyber-attacks with multiple transmission channels. *ISA Transactions.* 2021. P. 12–18. DOI: 10.1016/j.isatra.2021.11.033.

REFERENCES

1. Buriachok V.L., Kyrychok R.V., Skladannyi P.M. (2019). Osnovy informatsiinoi ta kibernetychnoi bezpeky. *Kyivskyi Universytet Imeni Borysa Hrinchenka.* https://elibrary.kubg.edu.ua/id/eprint/27370/.
2. Bezpeka informatsii v kompiuternykh systemakh ta merezhakh obiektiv krytychnoi infrastruktury: monohrafiia (2021) / S.F. Honchar, M.Iu. Komarov. Kyiv: *Try K*, 119.
3. Bilko S.S. (2023) Formuvannia informatsiinoi bezpeky natsionalnoi ekonomiky : dys. … dokt. filosofii : 051. *Natsionalnyi universytet "Poltavska politekhnika imeni Yuriia Kondratiuka"*. Poltava. 212.
4. Zastosuvannia veivlet-analizu dlia vyiavlennia anomalii merezhevoho trafiku. (2020). B.V. Petryk, V.R. Deineha, H.V. Nelasa. Suchasni problemy i dosiahnennia v haluzi radiotekhniky, telekomunikatsii ta informatsiinykh tekhnolohii: *Tezy dopovidei Kh Mizhnarodnoi naukovo-praktychnoi konferentsii, 07–09 zhovtnia 2020 r., m. Zaporizhzhia* / S.V. Morshchavka (vidpov. red.) Zaporizhzhia : NU "Zaporizka politekhnika". 163–167.
5. Metod vyiavlennia kiberatak na osnovi sotsialnoi inzhenerii. (2023). O.O. Bokhonko, S.M. Lysenko. *Zbirnyk naukovykh prats za materialamy XV Vseukrainskoi naukovo-praktychnoi konferentsii "Aktualni problemy kompiuternykh nauk APKN-2023"*. Khmelnytskyi, 38–41.
6. Rawther S., (2023). Entropy of a Computer Network Under Propagation of Cyber-Attacks.*International Journal of Engineering Trends and Technology*, *71*(8), 295–303. doi:10.14445/22315381/ijett-v71i8p226
7. Salih A.A., Abdulrazzaq M.B. (2023). Cyber security: performance analysis and challenges for cyber attacks detection.*Indonesian Journal of Electrical Engineering and Computer Science*, *31* (3), 1763. doi:10.11591/ijeecs.v31.i3.pp1763-1775.
8. Shafei Hamidreza, Li Li, Aguilera Ricardo. (2023). A Comprehensive Review on Cyber-Attack Detection and Control of Microgrid Systems. DOI: 10.1007/978-3-031-20360-2_1.
9. Spence Aaron, Bangay Shaun. (2022). Security beyond cybersecurity: side-channel attacks against non-cyber systems and their countermeasures. *International Journal of Information Security*. 21. DOI: 10.1007/s10207-021-00563-6.
10. Liu Xiao-Hui, Dong Jiuxiang, Yang Guang-Hong. (2023). Optimal DoS Attack Scheduling for Cyber-Physical Systems With Channel Hopping Scheme. *IEEE Transactions on Systems, Man, and Cybernetics: Systems.* 1–8. DOI: 10.1109/TSMC.2023.3305287.
11. Takahashi Makoto, Matsumoto Kohei. (2023). Experimental Study On The Cyber Attack Early Recognition System. *The Proceedings of the International Conference on Nuclear Engineering (ICONE)*. 30. 18–30. DOI:10.1299/jsmeicone.2023.30.1830.
12. Rawther Shiju, Sivaji Sathyalakshmi. (2023). Analysing the Spread of Cyber-Attacks in Computer Networks: A Simulation Study. *International Journal of Engineering Trends and Technology*. 71. 26–38. DOI: 10.14445/22315381/IJETT-V71I8P203.
13. Rajan S., Karthika R. (2022). A Survey of Computational Intelligence Techniques Used for Cyber-Attack Detection. DOI: 10.1007/978-981-19-3571-8_49.
14. Tahoun Ali, Arafa Mohammad. (2021). Secure control design for nonlinear cyber–physical systems under DoS, replay, and deception cyber-attacks with multiple transmission channels. *ISA Transactions*. 12–18. DOI: 10.1016/j.isatra.2021.11.033.