Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки
Назва теми

Рівень вищої освіти *другий (магістерський)*

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *123 «Комп'ютерна інженерія»*
Шифр, назва

Освітня програма *«Комп'ютерна інженерія та програмування»*
Назва

Шифр КвРКІ. 240242.24.02.25 ПЗ

Виконав здобувач IV курсу, група КІ2м-24-2 _____ Дмитро КУШНІР
Підпис Ініціали, прізвище

Керівник доктор філософії _____ Богдан САВЕНКО
Науковий ступінь, учене звання Підпис Ініціали, прізвище

Нормоконтролер канд.фіз.-мат.наук, доцент _____ Тетяна КИСІЛЬ
Науковий ступінь, учене звання Підпис Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІІС _____ Ольга ПАВЛОВА
«1» _травня_ 2026 р. підпис Ініціали, прізвище
дата

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ДРУГИЙ (МАГІСТЕРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ
Завідувачка кафедри КІІС
_____ Ольга ПАВЛОВА

" 12 " 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Кушніру Дмитру Віталійовичу
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки
Керівник проекту (роботи) Богдан САВЕНКО, доктор філософії
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання
Затверджена наказом ректора університету від 06.01.2026 №6
2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.
3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Аналіз відомих методів та засобів забезпечення функціонування розподілених систем на основі автоматизації критеріїв безпеки
Автоматизований аналіз впливу змін у комп'ютерних мережах на аргументи кібербезпеки
Формалізований каталог правил узгодженості та формальна модель взаємозв'язків між результатами перевірки і валідації безпеки
Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки, ефективність та експерименти
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтролер	Тетяна КИСІЛЬ, доцент каф. КІІС		
Антиплагіат	Андрій Нічепорук, доцент каф. КІІС		

7. Дата видачі завдання «_12_»_____01_____2026р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	12.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	12.01.2026	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	20.01.2026	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.02.2026	виконано
5	Робота над науковою статтею	01.03.2026	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.03.2026	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.206	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2026	виконано
9	Попередній захист ДРМ	29.04.2026	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2026	

Здобувач_____ ___Дмитро КУШНІР__
Підпис Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи_____ ___Богдан САВЕНКО___
Підпис Ім'я, ПРІЗВИЩЕ

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: «Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки»

Автор роботи: Кушнір Дмитро Віталійович

Керівник роботи: Савенко Б.О.

Пояснювальна записка: 81 с., 6 рис., 2 табл., 4 дод., 81 джерело.

АВТОМАТИЗАЦІЯ, ЗАХИСТ ПЕРИМЕТРУ, КІБЕРБЕЗПЕКА, КОМП'ЮТЕРНА МЕРЕЖА, КОМП'ЮТЕРНА СИСТЕМА, КОНТРОЛЬ ДОСТУПУ, КРИТЕРІЇ БЕЗПЕКИ, ПОКАЗНИКИ КОМПРОМЕТАЦІЇ, РОЗПОДІЛЕНА СИСТЕМА.

Об'єктом дослідження є процеси функціонування та автоматизації управління безпекою у розподілених системах.

Предметом дослідження є методи, моделі та засоби автоматизованого застосування формалізованих критеріїв безпеки під час функціонування розподілених систем.

Метою кваліфікаційної роботи магістра є покращення автоматичного застосування та контролю критеріїв безпеки на всіх рівнях функціонування розподілених систем та взаємодії їх компонентів.

Для розв'язання поставлених задач використовувалися теорія розподілених систем, теорія комп'ютерних мереж, теорія множин і теорія графів, також методи системного аналізу.

Наукова новизна отриманих результатів:

– новий метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки, який, на відміну від відомих підходів, що базуються на статичному налаштуванні політик та ручному адмініструванні засобів захисту, забезпечує інтегроване та формалізоване управління безпекою на всіх рівнях взаємодії компонентів системи, надає системний механізм керованості, передбачуваності та стійкості розподіленої інфраструктури за рахунок вбудованої, автоматизованої та формалізованої моделі

забезпечення безпеки, що функціонує в реальному часі та масштабується разом із системою.

Практична значимість отриманих результатів полягає у можливості застосування розробленого методу для підвищення рівня захищеності, керованості та стійкості функціонування розподілених систем різного призначення. Розроблений метод дозволяє автоматизувати процес застосування політик безпеки без необхідності постійного ручного адміністрування, забезпечити безперервний контроль відповідності функціонування системи формалізованим критеріям безпеки, зменшити ризик виникнення інцидентів, спричинених помилками конфігурації, скоротити час виявлення та локалізації порушень політик доступу, мінімізувати витрати на супровід та аудит безпеки, забезпечити масштабованість безпекових механізмів при зростанні кількості вузлів і сервісів. Практична реалізація методу можлива у середовищах з мікросервісною архітектурою, контейнеризованих платформах, хмарних інфраструктурах та системах з підвищеними вимогами до інформаційної безпеки. Отримані результати можуть бути використані при проєктуванні нових розподілених систем, модернізації існуючих IT-інфраструктур, а також під час впровадження концепцій DevSecOps та Zero Trust.

У вступі подано об’єкт та предмет дослідження, мету, наукову новизну та практичну цінність роботи, а також характеристику структури роботи.

У першому розділі проведено аналіз відомих рішень щодо стійкості розподілених систем, критеріїх їх безпеки та засобів автоматизації безпеки.

У другому розділі здійснено розроблення метамоделі аргументів безпеки.

У третьому розділі розроблено формальну модель взаємозв’язків між результатами перевірки та валідації безпеки.

У четвертому розділі здійснено розроблення методу, прогармного забезпечення, експериментів та оцінювання ефективності прийнятих рішень.

У висновках підведено підсумки досягнення результатів з розв’язання завдань дослідження.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

КС – Комп’ютерна система

ІТ – Інформаційні технології

ПЗ – Програмне забезпечення

GSN – Goal Structuring Notatio

ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується стрімким переходом до розподілених обчислювальних архітектур, зокрема хмарних платформ, мікросервісних середовищ, контейнеризованих інфраструктур та edge-обчислень. Такі системи забезпечують масштабованість, відмовостійкість та гнучкість, однак водночас істотно ускладнюють забезпечення інформаційної безпеки.

На відміну від централізованих систем, розподілені середовища мають динамічну топологію, велику кількість взаємодіючих вузлів, асинхронні комунікації та різнорівневу довіру між компонентами. Це призводить до зростання кількості потенційних вразливостей, ускладнення контролю доступу, моніторингу подій безпеки та своєчасного реагування на інциденти.

Традиційні підходи до забезпечення безпеки, що базуються на ручному адмініструванні політик і статичних правилах контролю доступу, виявляються недостатньо ефективними в умовах динамічних розподілених систем. Людський фактор, складність конфігурацій та масштаб інфраструктури підвищують ризик помилок і невідповідностей політик безпеки.

Актуальним напрямом досліджень є розроблення методів організації функціонування розподілених систем, що передбачають автоматизоване застосування формалізованих критеріїв безпеки. Такий підхід дозволяє інтегрувати вимоги безпеки безпосередньо в архітектуру системи, забезпечити їх постійний контроль, автоматичну верифікацію та адаптивне коригування.

Автоматизація застосування критеріїв безпеки передбачає формалізацію політик доступу, використання моделей довіри, механізмів контролю взаємодій між компонентами, а також впровадження інструментів моніторингу та оркестрації безпекових процесів.

Таким чином, розроблення методу організації функціонування розподілених систем на основі автоматизованого застосування критеріїв безпеки є актуальною науково-практичною задачею, спрямованою на підвищення стійкості, надійності та

керованості сучасних інформаційних інфраструктур.

Метою кваліфікаційної роботи є покращення автоматичного застосування та контролю критеріїв безпеки на всіх рівнях функціонування розподілених систем та взаємодії їх компонентів.

Поставлена мета досягається розв’язанням таких основних завдань:

- проаналізувати сучасні архітектури розподілених систем та існуючі підходи до забезпечення їх безпеки;

- визначити сукупність критеріїв безпеки, релевантних для розподіленого середовища (конфіденційність, цілісність, доступність, автентичність, невідмовність тощо);

- розробити формальну модель представлення критеріїв безпеки та політик їх реалізації;

- розробити метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки;

- розробити засіб з реалізованим в ній методом, провести експериментальну перевірку методу та оцінити ефективність методу за показниками зменшення кількості порушень політик, часу реагування на інциденти, обчислювальних витрат та масштабованості.

Об’єктом дослідження є процеси функціонування та автоматизації управління безпекою у розподілених системах.

Предметом дослідження є методи, моделі та засоби автоматизованого застосування формалізованих критеріїв безпеки під час функціонування розподілених систем.

Наукова новизна отриманих результатів:

- новий метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки, який, на відміну від відомих підходів, що базуються на статичному налаштуванні політик та ручному адмініструванні засобів захисту, забезпечує інтегроване та формалізоване управління безпекою на всіх рівнях взаємодії компонентів системи, надає системний механізм керованості, передбачуваності та стійкості розподіленої

інфраструктури за рахунок вбудованої, автоматизованої та формалізованої моделі забезпечення безпеки, що функціонує в реальному часі та масштабується разом із системою.

Практична значимість отриманих результатів полягає у можливості застосування розробленого методу для підвищення рівня захищеності, керованості та стійкості функціонування розподілених систем різного призначення. Розроблений метод дозволяє автоматизувати процес застосування політик безпеки без необхідності постійного ручного адміністрування, забезпечити безперервний контроль відповідності функціонування системи формалізованим критеріям безпеки, зменшити ризик виникнення інцидентів, спричинених помилками конфігурації, скоротити час виявлення та локалізації порушень політик доступу, мінімізувати витрати на супровід та аудит безпеки, забезпечити масштабованість безпекових механізмів при зростанні кількості вузлів і сервісів. Практична реалізація методу можлива у середовищах з мікросервісною архітектурою, контейнеризованих платформах, хмарних інфраструктурах та системах з підвищеними вимогами до інформаційної безпеки. Отримані результати можуть бути використані при проєктуванні нових розподілених систем, модернізації існуючих ІТ–інфраструктур, а також під час впровадження концепцій DevSecOps та Zero Trust.

Для розв’язання поставлених задач використовувалися теорія розподілених систем, теорія комп’ютерних мереж, теорія множин і теорія графів, також методи системного аналізу.

За темою кваліфікаційної роботи опубліковано одну статтю [81] у фаховому науковому журналі категорії Б «Вимірювальна та обчислювальна техніка в технологічних процесах» (№ 2, 2026).

1 АНАЛІЗ ВІДОМИХ МЕТОДІВ ТА ЗАСОБІВ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНУВАННЯ РОЗПОДІЛЕНИХ СИСТЕМ НА ОСНОВІ АВТОМАТИЗАЦІЇ КРИТЕРІЇВ БЕЗПЕКИ

1.1 Огляд та поняття стійкості розподілених систем щодо безпеки

Стійкість розподілених систем у сучасній парадигмі трактується як багатовимірна інтегральна характеристика, що поєднує інформаційну безпеку, функційну надійність, відмовостійкість, адаптивність та керованість у динамічному середовищі. Вона визначає здатність системи зберігати коректність стану, неперервність виконання логічних операцій і допустимий рівень сервісу за умов дії внутрішніх збоїв, мережних порушень, навмисних атак та невизначеності часових параметрів. Разом із тим, незважаючи на значний обсяг досліджень у галузі теорії розподілених обчислень, проблема комплексного забезпечення стійкості залишається остаточно не розв’язаною, що зумовлено фундаментальними алгоритмічними обмеженнями та зростаючою складністю сучасних інфраструктур [1, 2].

Відомі підходи до забезпечення стійкості зосереджені переважно на окремих аспектах, які включають реплікації даних, досягнення консенсусу, резервування ресурсів або впровадженні криптографічних механізмів захисту. Алгоритми консенсусу дозволяють забезпечити узгодженість стану в умовах відмов частини вузлів; проте вони супроводжуються істотними витратами обчислювальних і мережних ресурсів та не вирішують проблему динамічного конфлікту політик безпеки. Теорія відмовостійкості пропонує моделі активного та пасивного резервування, але не гарантує автоматичної узгодженості безпекових критеріїв у масштабі всієї системи. Криптографічні механізми забезпечують цілісність і автентичність повідомлень, однак не усувають ризиків логічної неконсистентності або накопичення прихованих конфігураційних помилок [3, 4].

Фундаментальним обмеженням функційної стійкості залишається компроміс, окреслений теоремою CAP [5, 6], відповідно до якої в умовах мережного поділу неможливо одночасно гарантувати строгі властивості

узгодженості, доступності та толерантності до розділення мережі. Існуючі архітектурні рішення змушені обирати між послабленими моделями узгодженості або зниженням доступності сервісу, що створює простір для потенційних зловживань і порушень цілісності. Невирішеною залишається проблема [7, 8] формалізованого автоматичного вибору оптимального режиму функціонування залежно від поточного стану мережі та рівня загроз.

Окремої уваги потребує проблема самостабілізації розподілених систем. Хоча теоретичні моделі самостабілізуючих алгоритмів доводять можливість повернення системи до легітимного стану з довільної конфігурації, практична інтеграція таких механізмів у великомасштабні інфраструктури ускладнена через відсутність формалізованих критеріїв легітимності та складність глобальної верифікації стану. Залишається відкритим питання побудови універсального механізму, здатного автоматично ідентифікувати деградаційні процеси та ініціювати процедури реконфігурації без централізованого контролю [9, 10].

Невирішеною також є проблема латентних прихованих відмов, які не призводять до повної зупинки вузла, але поступово спотворюють результати обчислень або накопичують помилкові стани. Традиційні механізми моніторингу орієнтовані на явні відмови та перевищення порогових значень, тоді як складні багатовекторні атаки або логічні конфлікти політик можуть залишатися непоміченими тривалий час. Відсутність формальної моделі узгодженості безпекових критеріїв у масштабі всієї системи унеможливлює гарантоване виявлення таких аномалій [11, 12].

Складність забезпечення стійкості зростає в асинхронних мережах, де неможливо достовірно відрізнити відмову вузла від його тимчасової затримки. Це породжує ризики розгалуження стану, появи суперечливих транзакцій та втрати глобальної консистентності. Існуючі протоколи частково вирішують цю проблему шляхом введення тайм-аутів або обрання лідера, однак такі підходи залишаються вразливими до навмисних затримок та атак типу «відмова в обслуговуванні» [13, 14].

В роботах [15, 16] запропоновано підходи до забезпечення стійкості

розподіленої інфраструктури в умовах впливів комп’ютерних атак.

Таким чином, можна констатувати, що в сучасних дослідженнях вирішено низку часткових задач таких, як забезпечення реплікації стану, формалізацію алгоритмів консенсусу, побудову моделей відмовостійкості, застосування криптографічних засобів підтвердження цілісності та розроблення механізмів резервування. Водночас комплексна проблема [17, 18] інтеграції функційної стійкості та інформаційної безпеки в єдину формалізовану систему автоматичного управління залишається відкритою.

Перспективними дослідженнями [19, 20, 21] є такі, що полягають в обґрунтуванні підходу, за якого стійкість розглядається як динамічний процес розподіленого середовища, який підтримується автоматичним застосуванням формалізованих критеріїв безпеки до кожної взаємодії між компонентами системи. Запропонована концепція передбачає інтеграцію алгоритмічної надійності, політик доступу, механізмів виявлення конфліктів і процедур реконфігурації в єдину модель управління, що функціонує в реальному часі. У межах такого підходу потенційно можуть бути вирішені проблеми автоматичного узгодження політик, виявлення їх конфліктності, мінімізації ризику логічної неконсистентності, а також адаптивного вибору режиму функціонування системи залежно від рівня загроз та стану мережної інфраструктури. Це дозволяє перейти від реактивної моделі захисту до проактивної адаптивної архітектури, у якій стійкість забезпечується не лише надмірністю ресурсів, а й інтелектуальним керуванням поведінкою компонентів.

Стійкість розподілених систем [22, 23] у контексті сучасних досліджень розглядається як багатовимірна характеристика, що формується під впливом сукупності структурних, алгоритмічних, безпекових, ресурсних, часових та адаптивних параметрів. Вона відображає здатність системи зберігати коректність функціонування, узгодженість стану, допустимий рівень продуктивності та захищеність від зовнішніх і внутрішніх загроз за умов невизначеності середовища. На відміну від традиційного розуміння надійності, яке зосереджується переважно на відмовах апаратних компонентів, стійкість розподіленої системи охоплює також

логічну цілісність даних, узгодженість транзакцій, здатність до реконфігурації та інтегрованість механізмів інформаційної безпеки у процес функціонування.

Одним із ключових параметрів [24, 25], що визначають стійкість, є структурна надмірність. Вона передбачає наявність резервних вузлів, дубльованих каналів зв’язку та реплік даних, які дозволяють зберігати працездатність у разі відмови окремих компонентів. Однак сама по собі надмірність не гарантує стійкості, якщо вона не супроводжується алгоритмами координації та узгодження стану між вузлами. Тому важливим параметром є топологічна зв’язність мережі, що визначає мінімальну кількість елементів, втрата яких призводить до фрагментації системи. Чим вищий рівень зв’язності, тим менша ймовірність виникнення ізольованих сегментів, у яких можуть накопичуватися суперечливі стани або виконуватися транзакції без глобальної синхронізації.

Суттєвим чинником стійкості [26, 27] є відсутність єдиної точки відмови. Централізовані елементи управління або зберігання ключових даних створюють потенційні вузли критичної залежності, компрометація чи відмова яких може призвести до колапсу всієї системи. Децентралізований характер архітектури підвищує стійкість за рахунок розподілу функцій управління, прийняття рішень і перевірки транзакцій між множиною учасників. Водночас надмірна децентралізація без формалізованих механізмів координації може спричинити втрату керованості, що підкреслює необхідність балансу між автономністю вузлів і централізованим контролем політик.

Алгоритмічна складова стійкості [28, 29] визначається параметрами, пов’язаними зі здатністю системи досягати узгодженого стану в умовах відмов або затримок. Важливим показником є максимальна кількість вузлів, відмова яких не порушує коректності роботи системи. Цей параметр залежить від обраної моделі відмов — чи передбачається лише зупинка вузлів, чи також їхня довільна або зловмисна поведінка. Алгоритми, здатні витримувати складні відмови, підвищують рівень стійкості, але потребують більших ресурсів і складніших процедур верифікації. Швидкість досягнення консенсусу між вузлами також є критичним параметром, оскільки затримки в узгодженні стану можуть призводити до

накопичення конфліктних транзакцій та підвищення ризику логічної нестабільності.

Параметри узгодженості даних [30, 31] безпосередньо впливають на стійкість. Строгі моделі консистентності забезпечують високий рівень коректності, проте знижують доступність у разі мережних поділів. Ослаблені моделі дозволяють підтримувати сервіс навіть за наявності тимчасових розбіжностей, але створюють ризик виникнення суперечливих станів. Стійкість у цьому контексті визначається здатністю системи обирати оптимальний режим узгодженості залежно від поточного стану інфраструктури та характеру навантаження.

Інформаційно-безпекові параметри [32, 33] є невід'ємною складовою стійкості. До них належить рівень формалізації політик доступу та контролю, який відображає, наскільки повно всі взаємодії між компонентами підлягають автоматичній перевірці відповідності визначеним критеріям. Чим більша частка операцій проходить через механізм формалізованого контролю, тим менша ймовірність несанкціонованих дій або конфігураційних помилок. Важливим є також ступінь узгодженості політик між різними підсистемами. Наявність суперечливих правил може створювати приховані вразливості або блокувати легітимні транзакції, що негативно впливає на загальну стабільність.

Здатність системи локалізувати інцидент без поширення його наслідків на інші компоненти є ще одним параметром стійкості [34, 35]. Механізми ізоляції, сегментації мережі та обмеження привілеїв дозволяють мінімізувати вплив компрометації окремого вузла. Важливими є також показники часу виявлення інциденту та часу відновлення після нього. Чим швидше система ідентифікує відхилення від нормального режиму та ініціює процедури реконфігурації, тим вищою є її стійкість. Водночас надмірна чутливість механізмів моніторингу може призводити до помилкових спрацьовувань, що створює додаткове навантаження.

Часові характеристики функціонування [36, 37] визначають здатність системи підтримувати прийнятну продуктивність навіть у деградованому режимі. Середній час між відмовами відображає довгострокову надійність компонентів, тоді як середній час відновлення демонструє ефективність процедур реагування.

Висока стійкість досягається тоді, коли інтервали стабільної роботи значно перевищують тривалість відновлення. Параметри затримки та пропускної здатності також мають значення, оскільки перевищення допустимих меж може призводити до каскадних збоїв.

Ресурсні параметри визначають запас міцності системи [38]. Наявність вільних обчислювальних потужностей, резервної пам’яті та пропускної здатності каналів дозволяє витримувати пікові навантаження або компенсувати втрату окремих вузлів. Нерівномірний розподіл навантаження створює точки потенційної деградації, тому важливим параметром є ефективність механізмів балансування. Система, здатна динамічно перерозподіляти запити та оптимізувати використання ресурсів, демонструє вищу стійкість у довгостроковій перспективі.

Адаптивність [39] є однією з визначальних характеристик сучасних розподілених систем. Вона включає здатність до автоматичної реконфігурації топології, зміни ролей вузлів, оновлення криптографічних ключів і перезапуску компонентів без зупинки сервісу. Параметри швидкості реконфігурації та частоти проактивного оновлення визначають, наскільки система готова до довготривалих атак або накопичення прихованих помилок. Самостабілізація, тобто здатність повертатися до легітимного стану без зовнішнього втручання, відображає рівень автономності механізмів підтримання стійкості.

Окремим параметром [40] є рівень інтеграції безпекових механізмів в алгоритмічну основу функціонування. Якщо безпека реалізована як зовнішній модуль, стійкість залежить від коректності взаємодії між підсистемами. У разі інтегрованого підходу критерії безпеки стають невід’ємною частиною логіки прийняття рішень, що підвищує передбачуваність поведінки системи. У такій моделі параметри безпеки безпосередньо впливають на алгоритми консенсусу, процедури обробки транзакцій та правила реконфігурації.

Важливим аспектом [41] є здатність системи функціонувати в умовах часткової деградації, забезпечуючи мінімально допустимий рівень сервісу. Параметри пріоритезації транзакцій і управління чергами дозволяють підтримувати виконання критичних операцій навіть за дефіциту ресурсів. Стійкість

у цьому випадку вимірюється не лише збереженням працездатності, а й контрольованим характером зниження продуктивності.

Узагальнюючи, стійкість розподіленої системи формується сукупністю взаємопов’язаних параметрів [42, 43], які охоплюють структуру мережі, алгоритми узгодження, рівень формалізації безпеки, часові характеристики, ресурсний запас і адаптивні можливості. Жоден із цих параметрів окремо не забезпечує повної стійкості; вона виникає як результат їхньої синергічної взаємодії. У контексті автоматичного застосування критеріїв безпеки стійкість набуває керованого характеру, оскільки параметри функціонування можуть коригуватися на основі формалізованих правил та аналітики стану системи. Це дозволяє розглядати стійкість не як статичну властивість, а як динамічний процес підтримання рівноваги між надійністю, продуктивністю та захищеністю в умовах постійної зміни середовища.

Отже, сучасний стан досліджень характеризується наявністю ефективних локальних рішень, але відсутністю універсальної методології, що поєднує алгоритмічну узгодженість, безпекові критерії та механізми самовідновлення в єдину формалізовану систему. Розроблення такого підходу становить актуальну наукову проблему та визначає напрям подальших досліджень у галузі організації функціонування стійких розподілених систем.

1.2 Аналіз критеріїв безпеки розподілених систем та засобів їх автоматизації

Стійкість [44, 45] розподілених систем є однією з ключових вимог до сучасних інформаційних інфраструктур, особливо в умовах постійних кіберзагроз, зростання навантаження та географічної розподіленості сервісів. Розподілена система за своєю природою складається з множини взаємопов’язаних вузлів, які можуть фізично перебувати у різних центрах обробки даних або навіть у різних країнах, але функціонують як єдине логічне ціле. Така архітектура забезпечує масштабованість і гнучкість, однак одночасно створює додаткові ризики: відмови

окремих компонентів; розсинхронізацію даних; перехоплення трафіку; атаки на канали зв’язку; компрометацію вузлів. У цьому контексті критерії інформаційної безпеки виступають не лише засобом захисту інформації, а й фундаментом забезпечення стійкості системи загалом. Саме через реалізацію цих критеріїв досягається здатність системи протистояти збоям і атакам, зберігати функціональність та відновлюватися без критичних наслідків.

Першим базовим критерієм є конфіденційність. Вона означає, що інформація доступна лише авторизованим суб’єктам і не може бути прочитана або скопійована сторонніми особами. У розподілених системах конфіденційність має особливе значення, оскільки дані постійно передаються між вузлами через мережу, а також можуть зберігатися у реплікованому вигляді на кількох серверах. Порушення конфіденційності одного вузла потенційно створює загрозу для всієї системи. Якщо зловмисник отримує доступ до внутрішнього трафіку або баз даних, він може не лише викрасти інформацію, а й використати її для подальшого розширення атаки. Тому забезпечення конфіденційності безпосередньо впливає на стійкість. Система, в якій дані зашифровані та ізольовані, здатна локалізувати наслідки інциденту й обмежити масштаб шкоди [44, 45].

Другим критерієм є цілісність. Вона гарантує, що дані залишаються точними, повними та незміненими без відповідних повноважень. У розподілених системах цілісність є складнішою проблемою, ніж у централізованих, оскільки дані синхронізуються між вузлами, передаються через мережу і можуть оброблятися паралельно. Порушення цілісності навіть на одному етапі може спричинити каскадні помилки, некоректні обчислення або втрату довіри до всієї системи. У критичних сферах, таких як банківські операції або державні реєстри, порушення цілісності фактично означає втрату працездатності системи. Отже, забезпечення цілісності прямо пов’язане зі стійкістю, оскільки тільки коректні та узгоджені дані дозволяють системі продовжувати функціонування після часткових збоїв [44, 45].

Третім критерієм є доступність. Він означає, що система та її ресурси залишаються доступними для користувачів у необхідний момент часу. У контексті стійкості доступність відіграє центральну роль. Розподілена система створюється

саме для підвищення доступності через реплікацію та резервування, однак без належного захисту вона може стати вразливою до атак типу відмови в обслуговуванні або перевантаження. Доступність визначає, чи здатна система виконувати свої функції навіть за умов часткової відмови компонентів або зовнішнього тиску. Втрата доступності означає фактичну втрату стійкості, незалежно від того, чи збережені конфіденційність і цілісність [44, 45].

Четвертим критерієм є автентичність. Вона передбачає можливість однозначно встановити особу користувача або вузла системи. У розподіленому середовищі, де вузли постійно обмінюються запитами та відповідями, автентичність захищає від впровадження фальшивих компонентів, підміни сервісів або атак типу «людина посередині». Якщо система не здатна перевіряти справжність учасників взаємодії, вона стає нестійкою, оскільки будь-який зловмисник може видавати себе за легітимний вузол і порушувати роботу інших компонентів.

П’ятим критерієм є невідмовність, яка забезпечує неможливість заперечення факту виконання певної дії. У розподілених системах це особливо важливо для аудиту, розслідування інцидентів та відновлення після атак. Наявність достовірних журналів подій дозволяє визначити причину збою, оцінити масштаби порушення та вжити коригувальних заходів. Без цього система стає неконтрольованою і втрачає здатність до адаптації [44, 45].

Розглянемо питання автоматизації критеріїв безпеки, оскільки саме автоматизація перетворює декларативні принципи безпеки на реальний інструмент забезпечення стійкості. Автоматизація конфіденційності передбачає автоматичне шифрування даних під час передачі та зберігання, автоматичне управління ключами, централізоване розгортання сертифікатів і контроль політик доступу. У сучасних системах шифрування каналів зв’язку між сервісами налаштовується за замовчуванням, а сертифікати можуть оновлюватися без участі адміністратора. Це дозволяє зменшити ризик людської помилки і забезпечити єдиний рівень захисту для всіх компонентів. Автоматизація ізоляції середовищ, сегментації мережі та контролю доступу робить систему більш стійкою до компрометації окремих

вузлів [45, 46].

Автоматизація забезпечення цілісності включає використання механізмів цифрового підпису, контрольних сум, автоматичного журналювання змін і перевірки узгодженості даних між вузлами. Системи можуть автоматично перевіряти коректність реплік, виявляти розбіжності та ініціювати процедури відновлення. Автоматизований контроль версій конфігурацій і коду дозволяє швидко повернутися до стабільного стану у разі помилки або атаки. Таким чином, цілісність підтримується не лише технічними механізмами, а й процесами безперервної перевірки та відновлення [47, 48].

Автоматизація доступності передбачає впровадження механізмів автоматичного масштабування, балансування навантаження, резервування ресурсів і самовідновлення сервісів. Автоматизація автентичності реалізується через централізовані служби ідентифікації, багатофакторну автентифікацію, автоматичне управління токенами доступу та інтеграцію сервісів за принципом взаємної довіри. Вузли системи можуть автоматично перевіряти сертифікати один одного перед встановленням з’єднання, що унеможливлює підключення неавторизованих компонентів. Політики доступу можуть оновлюватися централізовано і застосовуватися до всієї інфраструктури одночасно, що підвищує узгодженість та зменшує вразливості [49, 50].

Автоматизація невідмовності включає централізоване збирання логів, їх захищене зберігання, кореляцію подій та автоматичний аналіз інцидентів. Системи безпеки здатні виявляти підозрілу активність і ініціювати блокування або ізоляцію вузлів. Наявність повного та незмінного журналу подій дозволяє не лише розслідувати інциденти, а й вдосконалювати механізми захисту, що прямо впливає на довгострокову стійкість [51, 52].

Особливої уваги заслуговує інтегрована автоматизація всіх критеріїв одночасно. Сучасні підходи [53, 54] передбачають впровадження безпеки на всіх етапах життєвого циклу системи, від розробки до експлуатації. Автоматизовані процеси розгортання інфраструктури дозволяють створювати стандартизовані середовища з однаковими політиками безпеки. Моніторинг, реагування,

масштабування та відновлення працюють як єдиний механізм. У разі виявлення атаки система може автоматично ізолювати скомпрометований вузол, перерозподілити навантаження, відновити дані з резервної копії та оновити правила доступу. Такий комплексний підхід перетворює критерії безпеки на активний інструмент управління стійкістю.

Таким чином, критерії безпеки розподілених систем не є ізольованими вимогами [55, 56], а формують цілісну основу стійкості. Їх автоматизація дозволяє мінімізувати людський фактор, прискорити реагування на інциденти та забезпечити безперервність функціонування навіть в умовах складних кіберзагроз або технічних відмов. Саме поєднання глибокого розуміння критеріїв безпеки та впровадження автоматизованих механізмів їх реалізації створює передумови для побудови надійних, адаптивних і стійких розподілених систем, здатних ефективно функціонувати у сучасному цифровому середовищі.

Засоби забезпечення безпеки та стійкості розподілених систем становлять сукупність програмних, апаратних і організаційних інструментів, які реалізують критерії конфіденційності, цілісності, доступності, автентичності та невідмовності на практичному рівні. Якщо критерії визначають вимоги до системи, то засоби є механізмами їх досягнення та підтримки в автоматизованому режимі. У сучасних умовах саме інструментальні платформи й технологічні рішення формують основу стійкості, оскільки ручне адміністрування складних розподілених середовищ є неефективним і небезпечним через ризик людської помилки [57, 58].

До засобів забезпечення конфіденційності належать криптографічні механізми захисту даних під час передачі та зберігання. Основу становлять протоколи шифрування мережного трафіку, зокрема TLS, які автоматично встановлюють захищені канали між сервісами. У хмарних і корпоративних середовищах застосовується наскрізне шифрування між мікросервісами, що унеможливлює перехоплення внутрішнього трафіку навіть у разі компрометації окремого сегмента мережі. Додатково використовуються системи управління криптографічними ключами, які автоматично генерують, ротує та зберігають ключі в ізольованому середовищі. Важливим засобом є контроль доступу на основі ролей

і політик, що централізовано регулює права користувачів і сервісів. Такі механізми дозволяють обмежити поширення загрози в межах одного вузла та запобігти каскадному витоку даних, тим самим підвищуючи стійкість системи [59, 60].

Засоби забезпечення цілісності орієнтовані на виявлення та запобігання несанкціонованим змінам даних. До них належать механізми цифрового підпису, контрольні суми, хешування, а також системи перевірки узгодженості реплік у розподілених базах даних. У сучасних інфраструктурах широко застосовується контроль версій програмного коду й конфігурацій, що дозволяє швидко відновити попередній стабільний стан системи у разі помилки або атаки. Важливим засобом є автоматичне журналювання всіх змін із захистом логів від модифікації. Це забезпечує можливість відстеження джерела інциденту та швидке відновлення працездатності. Такі рішення формують основу довіри до даних і дозволяють системі продовжувати функціонування навіть після часткових збоїв [61].

Засоби забезпечення доступності спрямовані на підтримку безперервності роботи сервісів. До них належать балансувальники навантаження, механізми реплікації серверів, резервування каналів зв’язку, автоматичне масштабування ресурсів і кластери високої доступності [62].

Засоби забезпечення автентичності охоплюють системи керування ідентифікацією та доступом, служби каталогів, механізми багатофакторної автентифікації та взаємної перевірки вузлів. У розподілених архітектурах активно використовується принцип нульової довіри, за якого кожен запит до ресурсу проходить перевірку незалежно від розташування суб’єкта. Сертифікаційні центри автоматично видають і оновлюють цифрові сертифікати для сервісів, забезпечуючи їх взаємну довіру. Токени доступу мають обмежений термін дії та можуть відкликатися централізовано у разі виявлення загрози. Завдяки цим засобам система зменшує ризик проникнення несанкціонованих компонентів і зберігає контроль над внутрішніми процесами[63].

Засоби забезпечення невідмовності та аудиту реалізуються через централізовані системи збору логів і подій безпеки. Використовуються рішення класу SIEM, які аналізують потоки подій, виявляють аномалії та формують

сповіщення або автоматичні дії у відповідь. Логи зберігаються у захищеному середовищі з обмеженим доступом і механізмами контролю цілісності. У разі інциденту це дозволяє швидко встановити послідовність подій і визначити причину збою. Автоматизований аналіз журналів підвищує швидкість реагування та зменшує час простою системи [64].

Комплексне застосування зазначених засобів формує багаторівневу систему захисту, де кожен рівень підсилює інший. Криптографічні механізми захищають дані, системи моніторингу виявляють аномалії, оркестрація забезпечує самовідновлення, а аудит дозволяє аналізувати інциденти та вдосконалювати політики безпеки. У сукупності ці засоби створюють адаптивну екосистему, здатну автоматично реагувати на зміни та підтримувати безперервність функціонування навіть у складних умовах. Саме завдяки інтеграції технічних рішень, автоматизованих процедур і централізованого управління формується реальна стійкість розподілених систем, що відповідає сучасним вимогам кібербезпеки та надійності [65].

Засоби в контексті автоматизації використовуваних критеріїв безпеки в розподілених системах будемо розглядати як інтегровану сукупність технологій, інструментів і платформ, що забезпечують не тільки реалізацію окремих вимог конфіденційності, цілісності, доступності, автентичності та невідмовності, а їх постійну, автономну та узгоджену підтримку без критичної залежності від ручного адміністрування. У сучасних розподілених архітектурах обсяг компонентів, сервісів і взаємодій настільки великий, що забезпечення стійкості можливе лише за умови глибокої автоматизації кожного критерію безпеки та їх колективної взаємодії [66].

Автоматизація конфіденційності реалізується насамперед через засоби криптографічного захисту, які працюють у фоновому режимі та інтегровані на рівні інфраструктури. До них належать автоматичне шифрування каналів зв’язку між вузлами, автоматичне впровадження протоколів захищеної передачі даних, системи централізованого управління ключами та сертифікатами. У контейнеризованих і мікросервісних середовищах автоматизація конфіденційності

часто досягається шляхом впровадження сервісних мереж, які забезпечують шифрування трафіку між сервісами без необхідності зміни прикладного коду [671].

Автоматизація забезпечення цілісності базується на засобах контролю змін, перевірки достовірності даних і механізмах узгодження стану між вузлами. Сюди входять системи автоматичного контролю версій коду та конфігурацій, які відстежують кожну зміну і дозволяють швидко повернути систему до попереднього стабільного стану. У розподілених базах даних використовуються механізми автоматичної реплікації та перевірки узгодженості, що дозволяють виявляти розбіжності між копіями та ініціювати процедури синхронізації [68].

Автоматизація доступності охоплює широкий спектр засобів, що забезпечують безперервність функціонування навіть у разі часткових відмов. Основою є механізми кластеризації, балансування навантаження та автоматичного масштабування [69].

Автоматизація автентичності [69, 70, 71] реалізується через централізовані системи керування ідентифікацією та доступом, які інтегруються з усіма компонентами розподіленої системи. Кожен вузол або сервіс отримує унікальний ідентифікатор та цифровий сертифікат, який перевіряється під час кожної взаємодії. Механізми багатофакторної автентифікації для користувачів та взаємна TLS-автентифікація для сервісів дозволяють автоматично підтверджувати справжність сторін. У разі компрометації облікового запису або вузла доступ може бути відкликаний автоматично, що обмежує масштаби інциденту. Такий підхід [72, 73, 74] забезпечує сталість управління доступом навіть за умов масштабування системи.

Автоматизація невідмовності та аудиту [75, 76, 77] здійснюється через централізовані платформи збору й аналізу подій безпеки. Засоби класу SIEM акумулюють логи з усіх вузлів, виконують кореляцію подій та застосовують правила виявлення аномалій. У разі виявлення підозрілої активності система може автоматично ініціювати блокування облікового запису, ізоляцію вузла або зміну політики доступу. Захищене зберігання логів із контролем цілісності унеможливлює їх підробку, що забезпечує достовірність аудиту. Автоматизований

аналіз значно скорочує час реагування на інциденти та підвищує здатність системи до самовідновлення.

Отже, засоби автоматизації [78, 79, 80] використовуваних критеріїв безпеки у розподілених системах виступають ключовим механізмом забезпечення їх надійності та безперервності функціонування. Вони перетворюють концептуальні вимоги на конкретні технологічні рішення, які працюють постійно, масштабуються разом із системою та забезпечують швидке реагування на загрози. Саме інтеграція таких засобів дозволяє створити розподілену систему, здатну ефективно протистояти збоям, атакам і динамічним змінам навантаження без втрати контролю та керованості.

Стійкість розподілених систем безпосередньо залежить від системного впровадження критеріїв інформаційної безпеки та рівня їх автоматизації. Розподілене середовище за своєю природою є динамічним, масштабованим і потенційно вразливим до великої кількості внутрішніх і зовнішніх загроз, тому традиційні підходи до захисту, орієнтовані на ручне адміністрування або локальні механізми контролю, є недостатніми. Саме комплексне застосування критеріїв конфіденційності, цілісності, доступності, автентичності та невідмовності формує основу для забезпечення безперервності функціонування системи навіть в умовах часткових відмов або кібератак.

1.3 Постановка задачі

Поставлена мета досягається розв’язанням таких основних завдань:

- проаналізувати сучасні архітектури розподілених систем та існуючі підходи до забезпечення їх безпеки;

- визначити сукупність критеріїв безпеки, релевантних для розподіленого середовища (конфіденційність, цілісність, доступність, автентичність, невідмовність тощо);

- розробити формальну модель представлення критеріїв безпеки та політик їх реалізації;

- розробити метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки;

- розробити засіб з реалізованим в ній методом, провести експериментальну перевірку методу та оцінити ефективність методу за показниками зменшення кількості порушень політик, часу реагування на інциденти, обчислювальних витрат та масштабованості.

1.4 Висновки до першого розділу

Проаналізовано відомі методи та засоби забезпечення стійкості розподілених систем та автоматичного застосування критеріїв безпеки, які базуються на статичному налаштуванні політик та ручному адмініструванні засобів захисту..

2 АВТОМАТИЗОВАНИЙ АНАЛІЗ ВПЛИВУ ЗМІН У КОМП'ЮТЕРНИХ МЕРЕЖАХ НА АРГУМЕНТИ КІБЕРБЕЗПЕКИ

2.1 Метамодель аргументів безпеки

Для автоматизованого аналізу впливу змін у комп'ютерних мережах на аргументи кібербезпеки потрібно розробити метод, який забезпечуватиме підтримку актуальності та узгодженості доказів безпечності системи при зміні її конфігурації або структури. Для реалізації такого методу необхідно сформувати формалізовану модель представлення аргументів безпеки, що дозволить описувати взаємозв'язки між елементами аргументації, вимогами безпеки, доказами перевірки та компонентами мережної інфраструктури. У процесі дослідження необхідно систематизувати та сформувати каталог правил узгодженості між елементами аргументів безпеки та артефактами комп'ютерної мережі, такими як конфігурації вузлів, мережні служби, політики доступу, програмні модулі та результати перевірок безпеки. Ці правила повинні визначати, яким чином зміни у структурі або параметрах мережної системи впливають на відповідні елементи аргументації безпеки, а також дозволяти автоматично визначати елементи, які можуть втратити актуальність або потребують повторної перевірки. Крім того, необхідно визначити та реалізувати механізм семантичної простежуваності між моделями системної інженерії комп'ютерної мережі та моделями аргументів безпеки. Такий механізм повинен забезпечувати встановлення формальних зв'язків між елементами мережної інфраструктури та твердженнями аргументів безпеки, що дозволить автоматично відстежувати зміни у системі та визначати їх вплив на відповідні елементи аргументації. Це, у свою чергу, забезпечить можливість автоматизованого оновлення статусу аргументів безпеки при модифікації мережних компонентів або їх конфігурацій.

Важливим завданням дослідження є також формалізація взаємозв'язків між різними результатами перевірки та валідації безпеки, які використовуються як докази під час формування аргументів безпеки. Необхідно визначити правила взаємозалежності між різними типами доказів, зокрема результатами статичного

аналізу, тестування безпеки, перевірок конфігурацій та інших методів оцінювання захищеності. Це дозволить виконувати автоматизований аналіз повноти та достатності доказів безпечності системи, а також визначати вплив додавання, модифікації або видалення таких доказів на загальну структуру аргументації безпеки.

Для забезпечення практичної реалізації запропонованого підходу необхідно розробити структуру повторно використовуваних елементів аргументації безпеки, які можуть бути застосовані як типові шаблони або будівельні блоки під час формування аргументів безпеки комп'ютерних мереж. Для кожного такого елемента повинні бути визначені відповідні правила узгодженості та умови їх перевірки у разі зміни стану мережної системи.

Результатом виконання зазначених завдань має стати формування узагальненого методу організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки для автоматизованого аналізу впливу змін у комп'ютерних мережах на аргументи кібербезпеки, який забезпечуватиме можливість точного визначення порушень узгодженості між елементами аргументації безпеки, мережними артефактами та доказами перевірки безпеки, а також сприятиме підвищенню ефективності процесів управління змінами та підтримки актуальності аргументів безпеки у складних мережних системах.

Таким чином, потрібно здійснити розроблення метамоделі аргументів безпеки, множини типових шаблонів аргументів безпеки, механізм семантичної простежуваності кореляції між змінами, формалізований каталог правил узгодженості, формальна модель взаємозв'язків між результатами перевірки та валідації безпеки і метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки.

Спочатку розробимо метамодель аргументів безпеки сумісну зі стандартом GSN (Goal Structuring Notation), яка підтримує автоматизовану перевірку узгодженості та аналіз змін. Необхідно розробити метамодель аргументів безпеки, сумісну зі стандартом GSN, яка забезпечуватиме формалізоване представлення

структури аргументів кібербезпеки комп’ютерних мереж та підтримуватиме автоматизовану перевірку їх узгодженості. Запропонована метамодель повинна визначати основні елементи аргументації безпеки, зокрема цілі безпеки, стратегії доведення, твердження, контексти та докази, а також описувати формальні зв’язки між ними.

У межах розроблення метамоделі необхідно передбачити механізми встановлення прямих зв’язків між елементами аргументів безпеки та артефактами комп’ютерної мережі, такими як компоненти мережної інфраструктури, конфігураційні параметри, політики безпеки, результати перевірок і тестування. Це дозволить забезпечити семантичну простежуваність між моделями системної інженерії мережі та структурою аргументів безпеки.

Крім того, метамодель повинна підтримувати можливість анотування зв’язків між елементами аргументації та системними артефактами спеціалізованими правилами узгодженості, що визначають умови коректності аргументів безпеки у разі зміни параметрів або структури мережної системи. На основі таких правил має бути забезпечена автоматизована перевірка узгодженості аргументів безпеки та визначення їх актуальності після внесення змін до конфігурації комп’ютерної мережі.

Реалізація запропонованої метамоделі повинна створити основу для автоматизованого аналізу впливу змін у мережній інфраструктурі на аргументи кібербезпеки, що дозволить підвищити ефективність підтримки доказів безпечності системи та зменшити необхідність ручного аналізу під час управління змінами у комп’ютерних мережах. Відображення метамоделі діаграмою подано на рис. 2.1.

Наведена діаграма на рис. 2.1 відображає метамодель аргументів безпеки, побудовану відповідно до підходу Goal Structuring Notation, яка призначена для формалізації аргументів кібербезпеки та підтримки автоматизованого аналізу впливу змін у комп’ютерних мережах. Метамодель описує структуру аргументації безпеки, взаємозв’язки між її елементами, зв’язок з артефактами мережної системи, а також механізм перевірки узгодженості при зміні системних

компонентів.

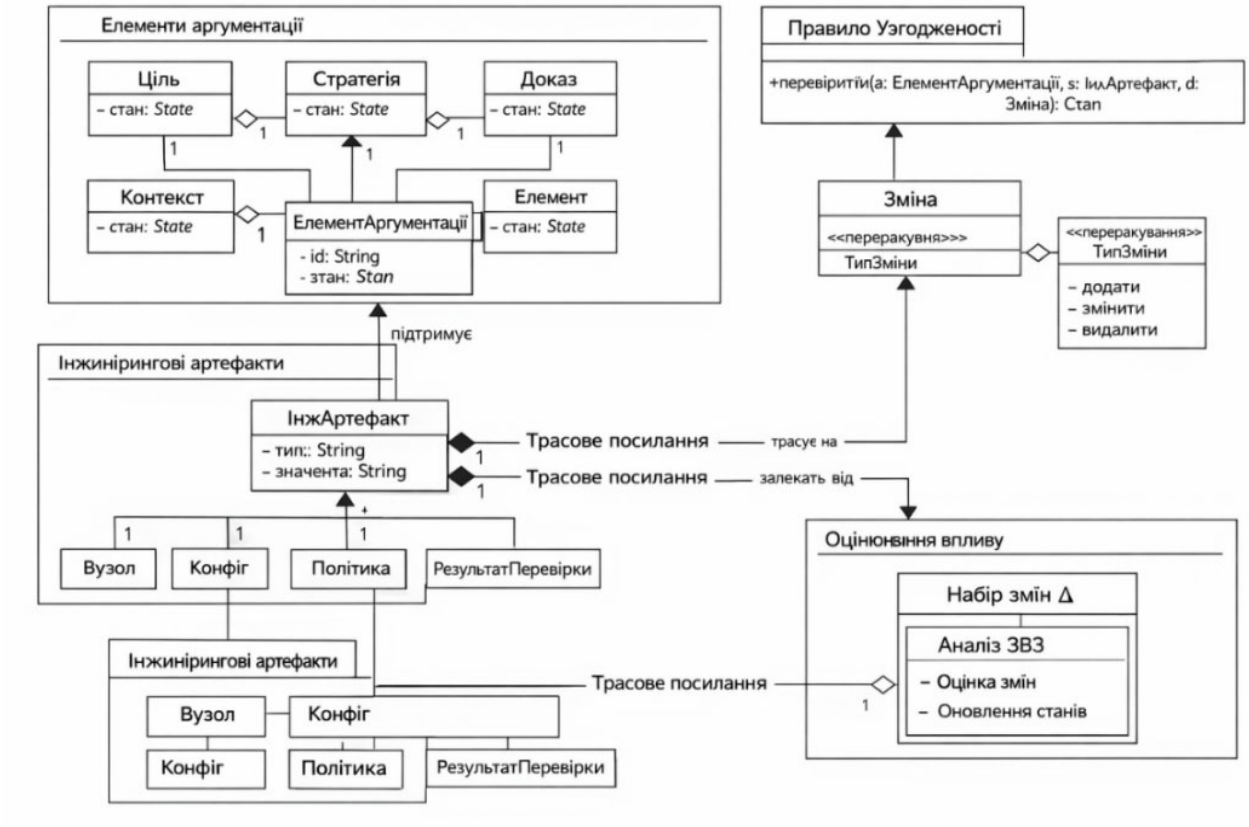


Рисунок 2.1 – Діаграма метамоделі

У верхній частині діаграми представлено елементи аргументації, які формують логічну структуру аргументів безпеки. Центральним елементом є клас Елемент аргументації, який узагальнює всі складові аргументу та містить основні атрибути, зокрема ідентифікатор та стан узгодженості. Від цього елемента наслідуються або логічно пов’язані основні типи компонентів аргументації: ціль; стратегія; доказ; контекст. Ціль визначає твердження щодо безпечності системи, яке необхідно довести. Стратегія описує спосіб або підхід, за допомогою якого здійснюється обґрунтування цієї цілі. Доказ містить результати перевірки або валідації, що підтверджують виконання відповідної вимоги безпеки. Контекст визначає умови, припущення або обмеження, у межах яких розглядається аргумент безпеки. Між цими елементами існують зв’язки підтримки, що формують ієрархічну структуру аргументації, де одна ціль може деталізуватися через стратегію, яка, у свою чергу, підкріплюється доказами.

У нижній лівій частині діаграми представлено інжинірингові артефакти системи, які відображають компоненти комп’ютерної мережі або результати її аналізу. Основним класом є ІнжАртефакт, який описує об’єкти системної інженерії та має атрибути типу і значення. До таких артефактів можуть належати мережні вузли, конфігурації системи, політики безпеки або результати перевірок і тестування. Ці артефакти є джерелами інформації, на основі яких формуються докази безпечності системи. Зв’язок між елементами аргументації та системними артефактами реалізується через трасові посилання, які забезпечують механізм простежуваності. Завдяки таким зв’язкам можна визначити, які саме компоненти системи впливають на конкретні елементи аргументів безпеки.

Праворуч на діаграмі показано модель змін у системі. Клас Зміна описує події модифікації системних артефактів, а перелік ТипЗміни визначає можливі види змін, зокрема додавання, модифікацію або видалення компонентів. Ці зміни можуть виникати під час оновлення програмного забезпечення, зміни конфігурації мережі або впровадження нових політик безпеки. Кожна така зміна може впливати на коректність аргументів безпеки, тому необхідно виконувати перевірку їх узгодженості.

Для цього у верхній правій частині діаграми введено елемент Правило узгодженості, який визначає функцію перевірки відповідності між елементом аргументації, системним артефактом та зміною, що відбулася. Це правило формалізує логіку визначення нового стану аргументу після внесення змін до системи. Результатом застосування правила є новий стан елемента аргументації, який може бути актуальним, неузгодженим або застарілим.

У нижній правій частині діаграми представлено механізм аналізу впливу змін, який реалізується у вигляді процесу оцінювання змін. Основним елементом є набір змін, що містить інформацію про всі модифікації системних артефактів. На основі цього набору виконується процедура аналізу впливу змін, яка включає оцінювання внесених змін та оновлення станів елементів аргументації. Завдяки цьому механізму система може автоматично визначати, які аргументи безпеки залишаються коректними, а які потребують повторної перевірки або перегляду.

Таким чином, представлена метамодель поєднує три ключові складові: структуру аргументів безпеки; модель системних артефактів комп'ютерної мережі; механізм аналізу змін. Така інтеграція дозволяє забезпечити семантичну простежуваність між аргументами безпеки та компонентами системи, а також автоматизувати процес перевірки узгодженості аргументації при зміні мережної інфраструктури. Використання запропонованої метамоделі створює основу для реалізації інструментальних засобів підтримки аргументів кібербезпеки та підвищує ефективність управління змінами у складних комп'ютерних мережах.

Метамодель аргументів безпеки представимо у вигляді формальної структури так:

$$M = (A, R, T, C), \tag{2.1}$$

де A - множина елементів аргументації безпеки;

R - множина відношень між елементами аргументації;

T - множина трасових зв'язків між аргументами безпеки та артефактами системи;

C - множина правил узгодженості.

Множину елементів аргументів безпеки визначимо так:

$$A = G \cup S \cup E \cup K, \tag{2.2}$$

де G - множина цілей безпеки (Goals);

S - множина стратегій доведення (Strategies);

E - множина доказів або результатів перевірок (Evidence);

K - множина контекстних елементів (Context).

Таким чином, задамо множину елементів аргументів безпеки так:

$$A = \{a_1, a_2, \dots, a_{n_A}\}, a_i = \left(a_{i,id}, a_{i,type}, a_{i,state}\right), \tag{2.3}$$

де a_i - i-тий елемент аргументації, що описується кортежем;

$i = 1,2, \dots, n_A$;

n_A – кількість елементів множини A;

$a_{i,id}$ - ідентифікатор елемента;

$a_{i,type}$ є $G \cup S \cup E \cup K$;

$a_{i,state}$ - стан узгодженості елемента.

Множина станів $A_{state} = \{a_{1,state}, a_{2,state}, \dots, a_{n_A,state}\}$, де $a_{i,state}$ є {дійсний, непослідовний, застарілий}.

Відношення аргументації визначимо множиною так:

$$B \subseteq A \times A, \tag{2.4}$$

де елемент $(a_i, a_j) \in B$ і означає, що він підтримує або деталізує елемент a_i.

Функцію підтримки для елементів з формули (2.4) визначимо так:

$$F_{support}: A \to 2^A, F_{support}(a_i) = \{a_j | (a_i, a_j) \in B\}, \tag{2.5}$$

де елемент $(a_i, a_j) \in B$ і означає, що він підтримує або деталізує елемент a_i;

B - відношення аргументації;

A - множина елементів аргументів безпеки;

2^A – множина підмножин множини A.

Модель системних артефактів комп'ютерної мережі задамо через артефакти системи множиною так:

$$S_{sys} = \{s_1, s_2, \dots, s_{n_{S_{sys}}}\}, \tag{2.6}$$

де $s_j = (s_{j,type}, s_{j,value})$;

$n_{S_{sys}}$ – кількість елементів множини S_{sys};

$j = 1,2, \dots, n_{S_{sys}}$.

Типи артефактів можуть включати вузли мережі, конфігурації, політики безпеки, результати тестування, програмні компоненти.

Зв'язок між аргументами безпеки та артефактами системи визначимо так:

$$T \subseteq A \times S_{sys}, \tag{2.7}$$

де елемент $(s_i, s_j) \in T$ і означає, що аргумент безпеки a_i залежить від артефакта системи s_j.

Функцію залежності визначимо так:

$$F_{trace}(a_i) = \{a_j | (a_i, s_j) \in T\}, \tag{2.8}$$

де елемент $(a_i, s_j) \in T$.

Зміни системи визначимо множиною так:

$$D_\Delta = \{(\delta, s_j) | \, j = 1,2, \dots, n_{S_{sys}}\}, \tag{2.9}$$

де s_j – змінений артефакт; δ - тип зміни;

$\delta \in \{\text{доповнено}, \text{видалено}, \text{модифіковано}\}$.

Правило узгодженості визначимо як функцію так:

$$c_k : A \times S_{sys} \times D_\Delta \to M_{state}, \tag{2.10}$$

де $M_{state} = \{\text{дійсний}, \text{непослідовний}, \text{застарілий}\}$;

M_{state} – множина станів;

D_Δ - множина змін системи;

A - множина елементів аргументів безпеки;

S_{sys} – множина артефактів системи.

Функція c_k визначає новий стан аргументу безпеки після зміни системного артефакта.

Стан елемента аргументації визначатимемо функцією так:

$$F_{state} = \begin{cases} \text{дійсний, якщо } c_k(a_i, s_j, \delta) = \text{дійсний;} \\ \text{непослідовний, якщо } c_k(a_i, s_j, \delta) = \text{непослідовний;} \\ \text{застаріший, якщо } c_k(a_i, s_j, \delta) = \text{застарілий,} \end{cases} \quad (2.11)$$

де $s_j = (s_{j,type}, s_{j,value})$;

$n_{S_{sys}}$ – кількість елементів множини S_{sys};

$j = 1,2, \dots, n_{S_{sys}}$;

a_i - i-тий елемент аргументації, що описується кортежем;

$i = 1,2, \dots, n_A$; n_A – кількість елементів множини A;

δ - тип зміни; $\delta \in \{\text{доповнено, видалено, модифіковано}\}$.

Аналіз впливу змін задамо функцією так:

$$F_v(M, D_\Delta) \to A', \quad (2.12)$$

де M - метамодель аргументів безпеки;

D_Δ - множина змін;

A' - оновлений стан аргументів.

Суть визначення функції F_v полягає у визначенні змінених артефактів s_j та знаходженні всіх аргументів, тобто формування множини так:

$$A_s = \{a_i | (a_i, s_j) \in T\}, \quad (2.13)$$

де елемент $(a_i, s_j) \in T$.

Формула (2.13) додатково доповнює формулу (2.8) і далі потрібно застосувати правила узгодженості з множини правил узгодженості C та оновити стан аргументів.

Таким чином, згідно введених понять та співвідношень (2.1)-(2.13) задамо концептуальну метамодель аргументів безпеки комп'ютерної мережі так:

$$M_k = (A, R, S_{sys}, T, C), \tag{2.14}$$

де A - множина елементів аргументації безпеки;

R - множина відношень між елементами аргументації;

S_{sys} - артефакти мережної системи;

T - множина трасових зв'язків між аргументами безпеки та артефактами системи;

C - множина правил узгодженості.

Запропонована метамодель аргументів безпеки, узгоджена з підходом Goal Structuring Notation, формує цілісну формалізовану основу для автоматизованої підтримки аргументів кібербезпеки у комп'ютерних мережах. Її ключова особливість полягає у поєднанні трьох раніше роз'єднаних компонентів: структури аргументації безпеки; моделей системних (мережних) артефактів; механізму аналізу впливу змін. На відміну від традиційного використання GSN як графічного засобу документування аргументів, запропонована метамодель розширює його до рівня формальної, машинозчитуваної структури з чітко визначеними трасовими зв'язками та правилами узгодженості.

Особливістю метамоделі є введення семантичної простежуваності між елементами аргументів безпеки та артефактами комп'ютерної мережі. Це дозволяє перейти від статичного опису аргументації до динамічної моделі, здатної реагувати на зміни конфігурації системи. У результаті аргументи безпеки перестають бути лише текстовими або графічними обґрунтуваннями і набувають властивостей формальної моделі, що підтримує автоматизовану перевірку узгодженості. Таким чином, метамодель забезпечує можливість мінімізувати ручний аналіз під час управління змінами, підвищити точність визначення впливу змін та зменшити

ризик використання застарілих доказів безпечності.

Практична цінність метамоделі полягає в тому, що вона дозволяє автоматично визначати аргументи безпеки, на які вплинули зміни у мережній інфраструктурі, формалізувати правила узгодженості між вимогами безпеки, доказами та системними компонентами, оцінювати повноту доказової бази після модифікації конфігурації системи, забезпечувати контроль актуальності аргументів безпеки в процесі експлуатації мережі. Таким чином, метамодель створює основу для побудови інтелектуальних інструментів підтримки процесів забезпечення кібербезпеки, аудиту та сертифікації мережних систем.

Отже, запропонована метамодель є теоретично обґрунтованою основою для автоматизованого аналізу впливу змін на аргументи кібербезпеки комп'ютерних мереж. Вона забезпечує формалізацію структури аргументації, інтеграцію з моделями системної інженерії та підтримку автоматизованої перевірки узгодженості. Подальше розширення та інструментальна реалізація метамоделі дозволять створити повноцінний метод підтримки актуальності аргументів безпеки у динамічних мережних середовищах.

2.2 Класи типових шаблонів аргументів безпеки для систем кіберзахисту комп'ютерних мереж

Одним із важливих елементів підтримки аргументів безпеки у складних інформаційних системах є використання типових шаблонів аргументації. У контексті розроблення аргументів безпеки для комп'ютерних мереж доцільно застосовувати підхід повторного використання структур аргументації, який передбачає формування набору стандартних елементів. Такі шаблони дозволяють значно спростити процес побудови аргументів безпеки, забезпечити їхню структурованість та підвищити узгодженість між різними компонентами аргументації. Формування подібних шаблонів ґрунтується на принципах, що застосовуються у підході Goal Structuring Notation, який дозволяє формалізувати аргументацію безпеки у вигляді ієрархічної структури цілей, стратегій та доказів.

Типові шаблони аргументів безпеки представляють собою формалізовані фрагменти аргументації, що відображають повторювані логічні структури обґрунтування безпечності системи. Кожний шаблон містить набір взаємопов’язаних елементів аргументації, які включають цілі безпеки, стратегії їх досягнення, контекстні припущення та відповідні докази. Такі шаблони створюються на основі аналізу типових вимог кібербезпеки та поширених механізмів захисту комп’ютерних мереж, зокрема механізмів контролю доступу, автентифікації, моніторингу мережного трафіку, виявлення вторгнень і захисту від несанкціонованих змін конфігурації системи.

Основною метою використання шаблонів аргументації є забезпечення повторного використання перевірених структур аргументів безпеки. Це дозволяє зменшити трудомісткість розроблення аргументів для нових систем, а також підвищити якість аргументації за рахунок використання раніше перевірених підходів. Крім того, застосування типових шаблонів сприяє уніфікації процесу побудови аргументів безпеки та спрощує їх аналіз і перевірку.

У межах запропонованого підходу шаблони аргументів безпеки розглядаються як модульні елементи метамоделі аргументації. Кожний шаблон визначає певну структуру зв’язків між елементами аргументації та містить заповнювачі для посилань на конкретні артефакти системної інженерії, такі як конфігурації мережного обладнання, політики безпеки або результати тестування. Під час застосування шаблону у конкретній системі ці заповнювачі заповнюються відповідними елементами мережної інфраструктури, що забезпечує інтеграцію аргументів безпеки з реальною моделлю системи.

Важливою характеристикою шаблонів є також можливість визначення правил узгодженості для кожного елемента аргументації. Такі правила описують умови, за яких аргумент залишається коректним, а також визначають можливі наслідки змін у відповідних системних артефактах. Завдяки цьому шаблони аргументів безпеки можуть використовуватися не лише для формування аргументації, але й для автоматизованого аналізу впливу змін у системі на її безпечність.

Таким чином, набір типових шаблонів аргументів безпеки є важливим компонентом метамоделі підтримки аргументів кібербезпеки комп’ютерних мереж. Використання таких шаблонів забезпечує повторне використання перевірених структур аргументації, підвищує узгодженість між елементами аргументів безпеки та системними артефактами, а також створює основу для автоматизованої перевірки їх актуальності у разі зміни конфігурації або структури мережної інфраструктури. Це, у свою чергу, сприяє підвищенню ефективності управління безпекою комп’ютерних мереж та підтримці актуальності доказів їхньої безпечності у динамічних умовах експлуатації.

У процесі формування аргументів безпеки для складних інформаційно-комунікаційних систем важливу роль відіграє використання типових структур аргументації, що дозволяють систематизувати обґрунтування безпечності системи та забезпечити повторне використання перевірених підходів. У контексті забезпечення кібербезпеки комп’ютерних мереж доцільним є формування набору типових шаблонів аргументів безпеки, які можуть застосовуватися як стандартні будівельні блоки під час розроблення аргументації. Такі шаблони формуються на основі підходів структурованої аргументації, що використовуються у методології Goal Structuring Notation, яка передбачає побудову аргументів безпеки у вигляді ієрархічної структури взаємопов’язаних цілей, стратегій, контекстів та доказів.

Типові шаблони аргументів безпеки представляють собою формалізовані фрагменти аргументації, які відображають повторювані логічні структури доведення безпечності системи або її окремих компонентів. Кожний такий шаблон містить набір елементів аргументації, що описують ціль безпеки, спосіб її досягнення, контекстні припущення та відповідні докази, які підтверджують виконання визначених вимог безпеки. Використання таких шаблонів дозволяє забезпечити уніфікацію процесу формування аргументів безпеки, зменшити трудомісткість їх розроблення, а також підвищити якість аргументації завдяки застосуванню перевірених структур обґрунтування.

Приклади типових шаблонів аргументів безпеки для комп’ютерних мереж.

1. Шаблон аргументу безпеки для контролю доступу використовується для

обґрунтування того, що доступ до ресурсів комп'ютерної мережі здійснюється лише авторизованими користувачами відповідно до встановленої політики безпеки.

2. Шаблон аргументу безпеки для захисту мережного периметра застосовується для підтвердження того, що мережа захищена від несанкціонованого доступу із зовнішніх мереж за допомогою міжмережних екранів та механізмів фільтрації трафіку.

3. Шаблон аргументу безпеки для виявлення вторгнень використовується для доведення здатності системи виявляти та реєструвати спроби несанкціонованого доступу або мережні атаки.

4. Шаблон аргументу безпеки для забезпечення цілісності конфігурації призначений для підтвердження того, що конфігурації мережних пристроїв та програмних компонентів не змінюються без контролю та відповідної авторизації.

Одним із прикладів типового шаблону є шаблон аргументу безпеки, що використовується для обґрунтування коректності механізмів контролю доступу до ресурсів комп'ютерної мережі. У цьому випадку основною ціллю аргументації є підтвердження того, що доступ до мережних ресурсів здійснюється виключно авторизованими користувачами відповідно до встановленої політики безпеки. Стратегія доведення такої цілі полягає у перевірці коректності механізмів автентифікації та авторизації, а також у підтвердженні відповідності реалізованих механізмів вимогам політики безпеки організації. Контекст аргументації визначається політиками контролю доступу, ролями користувачів та рівнями їхніх привілеїв у мережній системі. Доказами виконання вимог безпеки у цьому випадку можуть виступати конфігурації систем керування доступом, результати перевірки політик доступу, а також журнали автентифікації користувачів. Такий шаблон дозволяє сформувати узагальнену структуру аргументації, яка може бути застосована для різних мережних систем із подальшою деталізацією відповідних системних артефактів.

Іншим прикладом є шаблон аргументу безпеки для обґрунтування захисту мережного периметра. Основною ціллю такого шаблону є доведення того, що

мережа захищена від несанкціонованого доступу із зовнішніх джерел. У цьому випадку стратегія аргументації полягає у перевірці механізмів фільтрації мережного трафіку та коректності налаштувань міжмережних екранів. Контекст аргументації формується на основі політики мережної безпеки організації та переліку дозволених мережних служб. Доказами можуть виступати результати аналізу конфігурацій міжмережних екранів, результати сканування мережних портів, а також результати тестування на проникнення. Такий шаблон дозволяє формалізувати аргументацію безпеки, пов’язану із захистом мережної інфраструктури від зовнішніх загроз, і може використовуватися як типовий елемент у структурі аргументів безпеки.

Важливим прикладом також є шаблон аргументу безпеки для обґрунтування здатності системи виявляти спроби несанкціонованого доступу або мережних атак. У цьому випадку основною ціллю аргументації є підтвердження того, що система здатна своєчасно виявляти та реєструвати події безпеки, які можуть свідчити про наявність загроз. Стратегія доведення полягає у використанні механізмів моніторингу мережного трафіку, аналізу подій безпеки та систем виявлення вторгнень. Контекст аргументації визначається політикою моніторингу мережної інфраструктури та правилами аналізу подій безпеки. Доказами можуть виступати журнали роботи систем виявлення вторгнень, результати тестування механізмів виявлення атак, а також звіти систем моніторингу безпеки. Застосування такого шаблону дозволяє забезпечити структуроване обґрунтування ефективності механізмів виявлення кіберзагроз у мережній системі.

Ще одним важливим прикладом є шаблон аргументу безпеки для підтвердження цілісності конфігурації мережних компонентів. У цьому випадку основною ціллю аргументації є доведення того, що конфігурації мережних пристроїв та програмних компонентів не змінювалися без відповідного контролю та авторизації. Стратегія доведення полягає у використанні механізмів контролю змін, перевірки контрольних сум конфігураційних файлів та проведенні регулярних аудитів конфігурації. Контекст аргументації визначається політикою управління конфігураціями та регламентом внесення змін у систему. Доказами

можуть виступати журнали змін конфігурацій, результати перевірки контрольних сум, а також результати аудитів інформаційної безпеки.

Таким чином, використання типових шаблонів аргументів безпеки дозволяє сформувати стандартизований підхід до побудови аргументації кібербезпеки комп'ютерних мереж. Такі шаблони можуть використовуватися як повторно застосовувані структурні елементи аргументів безпеки та інтегруватися з моделями системної інженерії через механізми простежуваності. Це забезпечує можливість автоматизованого аналізу впливу змін у конфігурації мережної інфраструктури на аргументи безпеки, що є важливим елементом підтримки актуальності доказів безпечності системи у динамічних умовах експлуатації.

З часом типові шаблони аргументів безпеки для систем кіберзахисту комп'ютерних мереж можуть змінюватися під впливом розвитку технологій, зміни архітектур мережних систем, появи нових кіберзагроз та вдосконалення механізмів захисту. Такі зміни можуть стосуватися як структури самих шаблонів аргументації, так і змісту їхніх елементів, зокрема цілей безпеки, стратегій доведення, контекстних умов та доказів, що використовуються для підтвердження безпечності системи.

По-перше, зміни можуть відбуватися на рівні цілей безпеки, які визначають основні твердження аргументації. У міру розвитку комп'ютерних мереж і впровадження нових технологій можуть з'являтися додаткові вимоги до безпеки, наприклад вимоги до захисту хмарних сервісів, віртуалізованих середовищ або програмно-визначених мереж. У зв'язку з цим існуючі цілі аргументації можуть уточнюватися або доповнюватися новими підцілями, що відображають нові аспекти забезпечення кібербезпеки.

По-друге, зміни можуть стосуватися стратегій доведення безпеки, які визначають способи обґрунтування виконання вимог безпеки. Наприклад, із розвитком методів аналізу безпеки можуть з'являтися нові підходи до перевірки захищеності системи, такі як автоматизований аналіз конфігурацій, поведінковий аналіз мережного трафіку або застосування методів машинного навчання для виявлення аномалій. У такому випадку відповідні шаблони аргументів можуть бути

доповнені новими стратегіями або модифіковані для врахування нових методів аналізу.

По-третє, зміни можуть відбуватися у контексті аргументації, який визначає умови та припущення, у межах яких розглядається безпечність системи. Зокрема, можуть змінюватися політики інформаційної безпеки організації, регуляторні вимоги, стандарти безпеки або організаційні процедури управління мережною інфраструктурою. У результаті цього шаблони аргументів повинні бути адаптовані до нових умов експлуатації системи.

По-четверте, зміни можуть стосуватися доказів безпеки, які використовуються для підтвердження виконання вимог безпеки. З розвитком засобів моніторингу та аналізу мережних подій можуть змінюватися джерела доказів, наприклад додаватися результати автоматизованих сканувань вразливостей, результати тестування на проникнення, журнали систем виявлення вторгнень або звіти систем управління подіями безпеки. Водночас деякі види доказів можуть втрачати актуальність або замінюватися більш ефективними методами перевірки.

Крім того, зміни можуть відбуватися у зв’язках між шаблонами аргументації та системними артефактами комп’ютерної мережі. Наприклад, у разі модернізації мережної інфраструктури або зміни конфігурації мережних пристроїв можуть змінюватися відповідні елементи системної інженерії, з якими пов’язані аргументи безпеки. Це потребує оновлення трасових зв’язків між елементами аргументації та відповідними компонентами мережної системи.

Таким чином, типові шаблони аргументів безпеки не є статичними структурами, а повинні постійно адаптуватися до змін у технологіях, архітектурах мережних систем та вимогах до кібербезпеки. Регулярне оновлення таких шаблонів дозволяє підтримувати актуальність аргументів безпеки та забезпечує коректне відображення стану захищеності комп’ютерних мереж у процесі їх експлуатації та розвитку.

Зміни, що відбуваються у комп’ютерних мережах з часом, можуть безпосередньо впливати на коректність і актуальність аргументів безпеки,

сформованих на основі типових шаблонів. Оскільки аргументи безпеки базуються на конкретних системних артефактах (конфігураціях, політиках безпеки, програмних компонентах, результатах перевірок), то будь-яка модифікація цих елементів може змінювати обґрунтованість відповідних тверджень безпеки. У результаті деякі аргументи можуть залишатися коректними, інші потребуватимуть уточнення, а деякі можуть втратити актуальність.

У випадку шаблону аргументу безпеки для контролю доступу зміни можуть виникати через оновлення політики доступу, додавання нових користувачів або ролей, зміну механізмів автентифікації чи впровадження нових сервісів у мережі. Наприклад, якщо в систему додається новий мережний сервіс або змінюється структура ролей користувачів, то попередні аргументи безпеки, які підтверджували коректність контролю доступу, можуть більше не повністю відображати реальний стан системи. У такому випадку потрібно оновити доказову базу аргументу, зокрема перевірити конфігурації систем керування доступом, журнали автентифікації та відповідність нових налаштувань політиці безпеки.

Для шаблону аргументу безпеки щодо захисту мережного периметра зміни можуть бути пов’язані з модифікацією топології мережі, оновленням конфігурацій міжмережних екранів, додаванням нових мережних сегментів або впровадженням нових мережних сервісів. Наприклад, якщо відкриваються нові мережні порти або змінюються правила фільтрації трафіку, попередні аргументи безпеки, що підтверджували захищеність периметра, можуть втратити актуальність. У такому випадку необхідно повторно перевірити конфігурації мережного обладнання, результати сканування портів і результати тестування на проникнення, щоб переконатися, що нові зміни не створили додаткових вразливостей.

У випадку шаблону аргументу безпеки для виявлення вторгнень зміни можуть виникати через появу нових типів кіберзагроз, оновлення програмного забезпечення систем виявлення вторгнень або зміну правил аналізу мережного трафіку. Наприклад, якщо система виявлення вторгнень оновлюється або змінюються алгоритми аналізу подій безпеки, необхідно перевірити, чи зберігається здатність системи виявляти актуальні типи атак. У такому випадку

попередні докази безпеки, наприклад результати тестування або журнали подій, можуть вимагати оновлення або повторної перевірки.

Для шаблону аргументу безпеки щодо забезпечення цілісності конфігурації зміни можуть бути пов’язані з оновленням програмного забезпечення мережних пристроїв, зміною конфігураційних параметрів або впровадженням нових компонентів у мережну інфраструктуру. Наприклад, якщо змінюється конфігурація маршрутизаторів, серверів або систем захисту, необхідно перевірити, чи ці зміни були виконані відповідно до встановлених процедур управління конфігураціями. У протилежному випадку аргумент безпеки, що підтверджує цілісність конфігурації, може бути визнаний неузгодженим або застарілим.

Отже, зміни у часі можуть впливати на аргументи безпеки через модифікацію системних артефактів, на яких базується аргументація. Це означає, що аргументи безпеки повинні регулярно перевірятися на узгодженість із поточним станом мережної інфраструктури. Саме тому важливо використовувати формалізовані моделі аргументів безпеки та механізми аналізу впливу змін, які дозволяють автоматично визначати, які елементи аргументації залишаються актуальними, а які потребують оновлення або повторного підтвердження.

Для використання типових шаблонів аргументів безпеки у комп’ютерних мережах недостатньо лише їхнього текстового опису. Щоб такі шаблони могли бути використані в автоматизованих методах аналізу аргументів безпеки, їх необхідно подати у формалізованому вигляді. Формалізація дозволяє визначити структуру кожного шаблону, встановити зв’язки між його елементами та описати залежність аргументів безпеки від конкретних артефактів мережної системи. Далі введемо формальні моделі для чотирьох типових шаблонів аргументів безпеки комп’ютерних мереж.

Для обґрунтування безпечності системи контролю доступу здійснимо формальний опис твердження про те, що доступ до ресурсів мережі мають лише авторизовані користувачі. Це твердження буде враховувати конкретні політики безпеки та механізми автентифікації.

Задамо множину користувачів системи так:

$$U = \{u_1, u_2, \dots, u_{n_U}\}, \tag{2.15}$$

де u_i - i-тий користувач системи;

$i = 1,2, \dots, n_U$;

n_U – кількість користувачів системи в поточний момент часу.

Задамо множину ресурсів системи так:

$$Z = \{z_1, z_2, \dots, z_{n_Z}\}, \tag{2.16}$$

де z_i - i-тий ресурс системи;

$i = 1,2, \dots, n_Z$;

n_Z – кількість ресурсів системи в поточний момент часу.

Тоді політику доступу задамо відношенням так:

$$P_{pd} \subseteq U \times Z, \tag{2.17}$$

де пара (u_i, z_j) означає, що користувач u_i має право доступу до ресурсу z_j.

Аргумент безпеки контролю доступу задамо умовою так:

$$(u_i, z_j) \in A_{pd} \rightarrow (u_i, z_j) \in P_{pd}, \tag{2.18}$$

де A_{pd} - множина фактичних доступів до ресурсів системи.

Ця модель дозволяє формально перевіряти відповідність реальних дій користувачів встановленій політиці доступу. Якщо з часом змінюється політика P_{pd} або склад користувачів і ресурсів, то аргумент безпеки повинен бути переглянутий.

Для підтвердження захищеності мережного периметра необхідно формально

описати правила фільтрації мережного трафіку, що реалізуються міжмережними екранами. Формальну модель шаблону аргументу безпеки мережного периметра задамо так. Спочатку задамо множину мережних вузлів так:

$$H = \{h_1, h_2, \dots, h_{n_H}\}, \tag{2.19}$$

де h_i - i-тий мережний вузол системи;

$i = 1,2, \dots, n_H$;

n_H – кількість мережних вузлів системи в поточний момент часу.

Згідно формули (2.19) задамо множину можливих мережних з'єднань так:

$$M_{nc} \subseteq H \times H, \tag{2.20}$$

де пара (h_i, h_j) означає наявність мережного з'єднання та ці пари формують множину M_{nc}.

Множина дозволених з'єднань, яку визначено правилами міжмережного екрана, задамо так:

$$M_{nc,d} \subseteq M_{nc}, \tag{2.20}$$

де пара (h_i, h_j) є $M_{nc,d}$ означає наявність дозволеного мережного з'єднання та ці пари формують множину $M_{nc,d}$.

Тоді аргумент безпеки мережного периметра можна подати у вигляді умови з врахуванням формули (2.1) так:

$$c \in A; c \in C_{ext} \rightarrow c \notin M_{nc,d}, \tag{2.21}$$

де A - множина елементів аргументації безпеки;

C_{ext} – множина аргументів безпеки, що відображає з'єднання із зовнішніми

мережами, для випадку міжмережного екрану;

$M_{nc,d}$ – множина дозволених з'єднань.

Ця формалізація показує, що будь-яке несанкціоноване з'єднання із зовнішнього середовища повинно бути заблоковане правилами фільтрації. Якщо змінюються правила міжмережного екрана або топологія мережі, відповідний аргумент безпеки також потребує перевірки.

Для аргументації здатності системи виявляти атаки необхідно задати формально відповідність між подіями у мережі та механізмами їх виявлення. Нехай множину мережних подій задамо так:

$$P_{pd,n} = \left\{p_{pd,n,1}, p_{pd,n,2}, \dots, p_{pd,n,n_{P_{pd,n}}}\right\}, \tag{2.22}$$

де $p_{pd,n,i}$ - i-тий мережна подія;

$i = 1,2, \dots, n_{P_{pd,n}}$;

$n_{P_{pd,n}}$ – кількість мережних подій.

Множину подій, що відповідають атакам, задамо так:

$$P_{pd,n,a} \subseteq P_{pd,n}. \tag{2.23}$$

Система виявлення вторгнень формує множину зафіксованих інцидентів, яку задамо так:

$$P_{pd,n,ids} \subseteq P_{pd,n}. \tag{2.24}$$

Аргумент безпеки для виявлення вторгнень задамо умовою так:

$$e \in A; e \in P_{pd,n,a} \rightarrow e \in P_{pd,n,ids}. \tag{2.25}$$

Це означає, що кожна подія, яка відповідає атаці, повинна бути зафіксована системою виявлення вторгнень. Якщо з'являються нові типи атак або змінюються правила аналізу подій, то необхідно оновлювати відповідний аргумент безпеки.

Для доведення цілісності конфігурацій мережних пристроїв необхідно формально описати процес контролю змін конфігурації. Нехай множину конфігураційних параметрів мережних компонентів задамо так:

$$W = \{w_1, w_2, \dots, w_{n_H}\}, \tag{2.26}$$

де w_i - i-тий конфігураційниq параметр мережних компонентів;

$i = 1,2, \dots, n_W$;

n_W – кількість конфігураційних параметрів мережних компонентів.

Множину дозволених змін конфігурації задамо так:

$$W_{zkp} \subseteq W. \tag{2.27}$$

Множине фактичних змін конфігурації задамо так:

$$W_{fzkp} \subseteq W. \tag{2.28}$$

Аргумент безпеки цілісності конфігурації задамо умовою так:

$$w \in A; w \in W_{zkp} \rightarrow w \in W_{fzkp}. \tag{2.29}$$

Це означає, що будь-яка зміна конфігурації повинна бути санкціонована відповідними процедурами управління конфігураціями.

Наведені формальні моделі дозволяють представити типові шаблони аргументів безпеки комп'ютерних мереж у вигляді залежностей між елементами системи. Такий підхід забезпечує можливість формального аналізу коректності

аргументів безпеки та дозволяє автоматизувати перевірку їхньої актуальності у разі змін у конфігурації мережі, політиках безпеки або механізмах захисту. Формалізація шаблонів аргументації створює основу для побудови методу організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки.

2.3 Висновки до другого розділу

Запропонована метамодель аргументів кібербезпеки, узгоджена з підходом Goal Structuring Notation, забезпечує формалізовану основу для автоматизованої підтримки аргументів безпеки у комп'ютерних мережах. Її застосування дозволяє інтегрувати структуру аргументації безпеки з моделями мережних артефактів та механізмом аналізу впливу змін, що забезпечує семантичну простежуваність між вимогами безпеки, доказами та компонентами системи. Завдяки цьому аргументи безпеки переходять від статичного документування до динамічної формальної моделі, здатної автоматично реагувати на зміни конфігурації мережі. Реалізація такої метамоделі сприяє підвищенню точності аналізу впливу змін, зменшенню ризику використання неактуальних доказів безпечності та створює передумови для розроблення інтелектуальних інструментів підтримки процесів управління кібербезпекою, аудиту та сертифікації мережних систем.

Типові шаблони аргументів безпеки представляють собою формалізовані фрагменти аргументації, які відображають повторювані логічні структури доведення безпечності системи або її окремих компонентів. Кожний такий шаблон містить набір елементів аргументації, що описують ціль безпеки, спосіб її досягнення, контекстні припущення та відповідні докази, які підтверджують виконання визначених вимог безпеки. Використання таких шаблонів дозволяє забезпечити уніфікацію процесу формування аргументів безпеки, зменшити трудомісткість їх розроблення, а також підвищити якість аргументації завдяки застосуванню перевірених структур обґрунтування. Формалізація типових шаблонів аргументів безпеки додатково забезпечує можливість автоматизованої перевірки їхньої коректності та актуальності у динамічних мережних середовищах.

3 ФОРМАЛІЗОВАНИЙ КАТАЛОГ ПРАВИЛ УЗГОДЖЕНОСТІ ТА ФОРМАЛЬНА МОДЕЛЬ ВЗАЄМОЗВ'ЯЗКІВ МІЖ РЕЗУЛЬТАТАМИ ПЕРЕВІРКИ І ВАЛІДАЦІЇ БЕЗПЕКИ

3.1 Механізм семантичної простежуваності кореляції між змінами

У сучасних комп'ютерних мережах процес забезпечення кібербезпеки характеризується високою динамічністю, що пов'язано з постійними змінами у конфігураціях мережних пристроїв, оновленням програмного забезпечення, модифікацією політик доступу, появою нових сервісів та впровадженням додаткових механізмів захисту. Такі зміни є природною частиною експлуатації та розвитку мережної інфраструктури, проте вони можуть безпосередньо впливати на обґрунтованість раніше сформованих аргументів безпеки. Аргументи кібербезпеки формуються на основі конкретних припущень щодо структури системи, її конфігурацій, механізмів захисту та результатів перевірок безпеки. У разі зміни будь-якого з цих елементів може виникнути ситуація, коли частина аргументації втрачає актуальність або потребує повторного підтвердження.

Особливо складною є ситуація у великих або розподілених комп'ютерних мережах, де кількість взаємопов'язаних компонентів є значною, а зміни можуть відбуватися одночасно у різних частинах інфраструктури. У таких умовах ручний аналіз впливу змін на аргументи безпеки стає надзвичайно трудомістким і не завжди дозволяє своєчасно виявити потенційні порушення у структурі аргументації. Крім того, різні елементи аргументів безпеки можуть бути пов'язані з багатьма артефактами системної інженерії, такими як конфігурації мережних пристроїв, правила міжмережних екранів, політики доступу або результати тестування безпеки. Через це навіть незначні зміни у системі можуть мати складні та неочевидні наслідки для обґрунтованості аргументів кібербезпеки.

Для розв'язання цієї проблеми необхідно забезпечити можливість відстеження зв'язків між елементами аргументів безпеки та компонентами комп'ютерної мережі, на яких базується відповідна аргументація. Такий підхід дозволяє визначати, які саме твердження безпеки, стратегії доведення або докази

можуть бути затронуті у разі змін у системі. Важливою особливістю такого відстеження є не лише фіксація технічних залежностей між елементами різних моделей, але й врахування їхнього семантичного змісту. Іншими словами, необхідно встановлювати змістовні зв’язки між змінами у мережній інфраструктурі та відповідними елементами аргументації безпеки.

Саме з цією метою введемо механізм семантичної простежуваності кореляції між змінами. Його призначення полягає у формальному встановленні зв’язків між компонентами комп’ютерної мережі, результатами перевірки безпеки та елементами аргументів кібербезпеки. Такий механізм дозволяє відстежувати, які зміни у мережній системі можуть впливати на конкретні елементи аргументації, а також визначати, які твердження безпеки потребують повторної перевірки або оновлення. Наявність механізму семантичної простежуваності створює основу для автоматизованого аналізу впливу змін у комп’ютерних мережах на аргументи кібербезпеки та забезпечує підтримку їх актуальності у процесі експлуатації та розвитку мережних систем.

Для забезпечення актуальності аргументів безпеки у комп’ютерних мережах необхідно враховувати той факт, що мережні системи постійно змінюються. Зміни можуть стосуватися конфігурацій мережних пристроїв, політик доступу, програмного забезпечення, топології мережі або механізмів захисту. Оскільки аргументи безпеки базуються на конкретних артефактах системи та результатах перевірок безпеки, будь-яка зміна цих елементів може впливати на коректність відповідних тверджень безпеки. У зв’язку з цим виникає необхідність розроблення механізму семантичної простежуваності кореляції між змінами, який дозволяє встановлювати зв’язки між змінами у мережній системі та елементами аргументів безпеки, що можуть бути ними порушені.

Семантична простежуваність передбачає формальне встановлення зв’язків між елементами різних моделей: моделлю комп’ютерної мережі, моделлю аргументів безпеки та результатами перевірки й валідації системи. Для цього введемо множину елементів аргументації безпеки

$$N_a = N_g \cup N_s \cup N_e, \tag{3.1}$$

де N_g - множина цілей безпеки;

N_s - множина стратегій аргументації;

N_e - множина доказів безпеки.

Паралельно введемо множину елементів мережної системи так:

$$S_{net} = \left\{s_{net,1}, s_{net,2}, \dots, s_{net,n_{S_{net}}}\right\}, \tag{3.2}$$

де $s_{net,i}$ - i-та компонента мережі;

$i = 1,2, \dots, n_{S_{net}}$;

$n_{S_{net}}$ – кількість компонент в мережі.

Множина S_{net} містить компоненти комп'ютерної мережі, зокрема вузли, мережні служби, конфігурації пристроїв, правила міжмережних екранів та інші артефакти системної інженерії.

Для встановлення зв'язків між елементами аргументів безпеки та компонентами мережної системи введемо відношення семантичної простежуваності так:

$$T_{sp} \subseteq N_a \times S_{net}, \tag{3.3}$$

де пара $\left(n_i, s_{net,j}\right)$ означає що елемент аргументації n_i семантично залежить від елемента мережної системи $s_{net,j}$.

Наприклад, доказ безпеки може бути пов'язаний із конкретною конфігурацією міжмережного екрану або результатом тестування системи виявлення та запобігання вторгнень.

Однак для аналізу впливу змін необхідно формально описати самі зміни у мережній системі. Для цього введемо множину змін так:

$$Q_{net} = \left\{q_{net,1}, q_{net,2}, \dots, q_{net,n_{Q_{net}}}\right\}, \quad (3.4)$$

де $q_{net,i}$ - i-та зміна у мережній системі;

$i = 1,2, \dots, n_{Q_{net}}$;

$n_{Q_{net}}$ – кількість змін у мережній системі.

К кожна зміна $q_{net,i}$ відображає модифікацію певного елемента мережної системи. Формально кожну зміну подамо як відображення так:

$$q_{net,i}: s_{net,i} \rightarrow s'_{net,i}, \quad (3.5)$$

де $s_{net,i}$ - початковий стан елемента системи, а $s'_{net,i}$ - стан після зміни.

Після цього введемо відношення кореляції змін, яке дозволить встановити, які елементи аргументації можуть бути затронуті певною зміною, так:

$$R_{sp} \subseteq Q_{net} \times N_a, \quad (3.6)$$

де пара $\left(q_{net,i}, n_{a,i}\right)$ означає, що зміна $q_{net,i}$ потенційно впливає на елемент аргументації $n_{a,i}$.

Таке відношення для опису кореляції змін визначимо через відношення простежуваності так:

$$\left(q_{net,i}, n_{a,i}\right) \in R_{sp} \leftrightarrow \left(n_{a,i}, s_{net,j}\right) \in T_{sp}$$

де $s_{net,j}$ є елементом системи, що змінюється у результаті виконання зміни $q_{net,i}$.

Ця модель дозволяє визначити множину аргументів безпеки, які можуть бути порушені внаслідок змін у мережній системі. Для цього введемо функцію впливу змін

$$F_I(q_{net,i}) = \{n_{a,i} \in N_a | (q_{net,i}, n_{a,i}) \in R_{sp}\}, \quad (3.7)$$

де $F_I(q_{net,i})$ - множина елементів аргументації, на які впливає зміна $q_{net,i}$.

У процесі експлуатації комп'ютерних мереж аргументи безпеки формуються на основі конкретних технічних характеристик системи, конфігурацій мережних пристроїв, політик доступу, механізмів захисту та результатів перевірок безпеки. Проте мережні інфраструктури є динамічними системами, у яких постійно відбуваються зміни: оновлюється програмне забезпечення; змінюються правила доступу; додаються нові сервіси; модифікуються параметри мережних пристроїв; впроваджуються нові засоби захисту. У результаті таких змін виникає ризик втрати актуальності раніше сформованих аргументів безпеки, оскільки вони можуть базуватися на припущеннях або доказах, що вже не відповідають поточному стану системи.

Особливо важливо враховувати вплив змін на типові шаблони аргументів безпеки, які використовуються для обґрунтування захищеності різних аспектів функціонування комп'ютерних мереж. Кожний шаблон аргументації пов'язаний із певними компонентами мережної інфраструктури та відповідними механізмами захисту. Тому зміни у цих компонентах можуть безпосередньо впливати на коректність аргументації. У зв'язку з цим необхідно забезпечити механізм семантичної простежуваності, який дозволяє встановлювати змістовні зв'язки між змінами у мережній системі та відповідними елементами аргументів безпеки.

Зокрема, у випадку шаблону аргументу безпеки для контролю доступу аргументація базується на припущенні, що доступ до ресурсів комп'ютерної мережі здійснюється виключно відповідно до визначених політик доступу та процедур автентифікації користувачів. Однак у реальних системах можуть відбуватися зміни, пов'язані з додаванням нових користувачів, модифікацією ролей доступу, впровадженням нових сервісів або зміною механізмів автентифікації. Такі зміни можуть впливати на коректність твердження про те, що доступ до ресурсів мережі є належним чином контрольованим. Тому механізм семантичної простежуваності повинен забезпечувати можливість встановлення зв'язку між змінами у політиках

доступу або системах автентифікації та відповідними елементами аргументації безпеки.

Аналогічно, у випадку шаблону аргументу безпеки для захисту мережного периметра аргументація базується на припущенні, що мережа захищена від несанкціонованого доступу із зовнішнього середовища за допомогою міжмережних екранів та механізмів фільтрації трафіку. Проте зміни у конфігурації міжмережних екранів, відкриття нових мережних портів, додавання нових мережних сегментів або модифікація топології мережі можуть змінювати рівень захищеності периметра. У такій ситуації аргумент безпеки, який підтверджує захищеність мережного периметра, може втратити актуальність або потребувати повторної перевірки. Саме тому необхідно забезпечити можливість встановлення семантичних зв’язків між конфігураціями мережних засобів захисту та відповідними елементами аргументів безпеки.

У випадку шаблону аргументу безпеки для виявлення вторгнень аргументація ґрунтується на здатності системи виявляти та реєструвати мережні атаки або інші підозрілі події. Проте ефективність таких систем може змінюватися внаслідок оновлення програмного забезпечення, модифікації правил аналізу мережного трафіку або появи нових типів атак. Якщо система виявлення вторгнень не адаптована до нових загроз або її конфігурація була змінена, це може впливати на коректність аргументу безпеки щодо здатності системи своєчасно виявляти атаки. У зв’язку з цим необхідно забезпечити механізм відстеження зв’язків між змінами у системах моніторингу безпеки та відповідними доказами аргументації.

Подібна ситуація виникає і для шаблону аргументу безпеки щодо цілісності конфігурації мережних компонентів. Аргументація у цьому випадку базується на припущенні, що всі зміни конфігурацій мережних пристроїв або програмних компонентів виконуються відповідно до визначених процедур управління конфігураціями та проходять необхідні процедури авторизації. Проте у процесі експлуатації системи можуть відбуватися зміни конфігурацій, пов’язані з оновленням програмного забезпечення, виправленням вразливостей або модернізацією мережної інфраструктури. Якщо такі зміни виконуються без

належного контролю або не враховуються у структурі аргументації, відповідний аргумент безпеки може втратити достовірність.

Таким чином, для кожного із розглянутих шаблонів аргументів безпеки існує тісний зв’язок між елементами аргументації та конкретними компонентами комп’ютерної мережі. Зміни у цих компонентах можуть мати безпосередній вплив на коректність відповідних тверджень безпеки. Саме тому виникає необхідність у розробленні механізму семантичної простежуваності кореляції між змінами, який дозволяє встановлювати змістовні зв’язки між змінами у мережній інфраструктурі та відповідними елементами аргументів кібербезпеки. Реалізація такого механізму створює основу для автоматизованого аналізу впливу змін у комп’ютерних мережах на аргументи безпеки та забезпечує підтримку їх актуальності у процесі розвитку та експлуатації мережних систем.

Таким чином, розроблений механізм семантичної простежуваності дозволяє встановити формальні зв’язки між змінами у комп’ютерній мережі та відповідними елементами аргументів безпеки. Завдяки цьому стає можливим автоматизований аналіз впливу змін на коректність аргументів кібербезпеки. У разі виникнення змін система може автоматично визначити, які саме цілі, стратегії або докази аргументації потребують повторної перевірки або оновлення. Такий підхід забезпечує підтримку актуальності аргументів безпеки у динамічному середовищі експлуатації комп’ютерних мереж і створює основу для реалізації методів автоматизованого управління аргументами кібербезпеки.

3.2 Формалізований каталог правил узгодженості

У процесі розроблення метамоделі аргументів безпеки та механізмів їх автоматизованого аналізу важливо враховувати, що наявність формальної структури аргументації та механізму семантичної простежуваності сама по собі не гарантує коректності та достовірності аргументів кібербезпеки. Навіть за умови встановлення зв’язків між елементами аргументації та компонентами комп’ютерної мережі можуть виникати ситуації, коли аргументи є неповними, суперечливими або

не відповідають поточному стану системи. Це зумовлено тим, що у складних мережних середовищах різні елементи аргументації формуються на основі неоднорідних джерел інформації, які можуть змінюватися незалежно один від одного.

Крім того, у процесі еволюції комп'ютерної мережі відбуваються численні зміни, що можуть впливати на різні аспекти аргументації безпеки, тобто від оновлення конфігурацій мережних пристроїв до зміни політик доступу або результатів перевірки системи. У таких умовах навіть коректно побудований аргумент безпеки може з часом втратити свою узгодженість. Наприклад, окремі твердження можуть залишитися без належного обґрунтування, докази можуть втратити актуальність, або між різними елементами аргументації можуть виникнути логічні суперечності. З огляду на це виникає необхідність у введенні формалізованих правил, які б регламентували вимоги до коректності аргументів безпеки та дозволяли автоматично перевіряти їх узгодженість. Такі правила повинні охоплювати різні аспекти аргументації, включаючи її структуру, відповідність системним артефактам, актуальність у часі, логічну несуперечність та узгодженість із результатами перевірки і валідації безпеки. Важливою особливістю цих правил є те, що вони мають бути сформульовані у формалізованому вигляді, що забезпечуватиме можливість їх використання в автоматизованих методах аналізу аргументів кібербезпеки.

Таким чином, для забезпечення цілісності, повноти та актуальності аргументів безпеки доцільно сформувати формалізований каталог правил узгодженості, який визначає систему обмежень і умов, що повинні виконуватися для кожного елемента аргументації. Наявність такого каталогу дозволяє не лише виявляти порушення у структурі аргументів безпеки, але й забезпечує основу для їх автоматизованої перевірки та підтримки в умовах постійних змін у комп'ютерних мережах.

У процесі побудови та підтримки аргументів кібербезпеки для комп'ютерних мереж важливим завданням є забезпечення їх узгодженості як на рівні внутрішньої структури аргументації, так і на рівні відповідності елементам мережної системи

та результатам перевірки безпеки. В умовах динамічних змін мережної інфраструктури, модифікації політик безпеки та оновлення механізмів захисту виникає необхідність у формалізованому підході до контролю узгодженості аргументів безпеки. Такий підхід дозволяє своєчасно виявляти суперечності, неповноту або втрату актуальності аргументації.

Формалізований каталог правил узгодженості задамо впорядкованою множиною правил, що будуть регламентувати допустимі зв'язки між елементами аргументації, а також умови їх відповідності зовнішнім артефактам системи. Для формалізації такого каталогу введемо множину правил узгодженості так:

$$R_{pr} = \left\{ r_{pr,1}, r_{pr,2}, \dots, r_{pr,n_{R_{pr}}} \right\}, \tag{3.8}$$

де $r_{pr,i}$ - i-те правило;

$i = 1,2, \dots, n_{R_{pr}}$;

$n_{R_{pr}}$ – кількість правил.

Кожне правило $r_{pr,i}$ визначає певне обмеження або умову коректності аргументації безпеки.

Першу групу складають структурні правила узгодженості, які визначають допустимі зв'язки між елементами аргументації. Наприклад, кожна ціль безпеки повинна бути обґрунтована хоча б однією стратегією або набором доказів. Це правило подамо так:

$$\forall n_g \in N_g \; \exists n_a \in (N_s \cup N_e) : \left(n_g, n_a\right) \in R_v, \tag{3.9}$$

де N_g - множина цілей безпеки;

N_s - множина стратегій аргументації;

N_e - множина доказів безпеки;

R_v - множина відношень у структурі аргументації.

Запровадження цього правила дозволяє гарантувати відсутність

необґрунтованих тверджень у структурі аргументу безпеки. Проте цього недостатньо, оскільки важливо також забезпечити коректність самих доказів.

До другої групи правил віднесемо правила узгодженості доказів, які визначають відповідність доказів реальним артефактам системи. Кожен доказ повинен бути пов’язаний із конкретним джерелом у мережній системі і задамо його так:

$$\forall n_e \in N_e \; \exists a_a \in A_a : F_T(n_e) = a_a, \tag{3.10}$$

де A_a - множина системних артефактів;

F_T - функція трасування.

Це правило забезпечує можливість перевірки достовірності доказів. Водночас у динамічному середовищі важливо враховувати, що артефакти системи можуть змінюватися, що потребує введення додаткових правил.

До третьої групи віднесемо правила часової узгодженості, які враховують актуальність доказів у часі. Зокрема, кожен доказ повинен відповідати поточному стану системи і задамо його так:

$$F_{time}(n_e) \geq F_{time}(a_a), \tag{3.11}$$

де F_{time} – функція отримання часу;

$F_{time}(n_e)$ - час отримання доказу;

$F_{time}(a_a)$ - час останньої зміни відповідного артефакту.

Це правило дозволяє виявляти застарілі докази, що втратили актуальність унаслідок змін у системі. Однак навіть актуальні докази повинні бути узгоджені між собою.

До четвертої групи віднесемо правила логічної узгодженості, які забезпечують відсутність суперечностей у структурі аргументації. Наприклад, не повинно існувати двох доказів, що приводять до взаємовиключних висновків. Задамо ціправила так:

$$\nexists\, n_{e,1}, n_{e,2} \in N_e : F_{conflict}(n_{e,1}, n_{e,2}) = true, \quad (3.12)$$

де функція $F_{conflict}$ визначає наявність суперечності між доказами.

Ці правила дозволяють забезпечити логічну цілісність аргументації. Проте для комплексного аналізу необхідно також враховувати зв'язки між аргументами безпеки та результатами перевірки системи.

До п'ятої групи правил віднесемо правила узгодженості з результатами перевірки та валідації, які визначають відповідність аргументів безпеки результатам тестування, аудиту або моніторингу системи і задамо їх так:

$$\forall n_g \in N_g : F_{valid}(n_g) \rightarrow \exists v_s \in V_s, \quad (3.13)$$

де V_s - множина результатів перевірки безпеки.

Це правило гарантує, що кожне твердження безпеки має підтвердження у вигляді результатів перевірки або валідації. Водночас необхідно враховувати, що результати перевірки також можуть змінюватися.

Таким чином, формалізований каталог правил узгодженості забезпечує комплексний контроль коректності аргументів безпеки, включаючи їх структуру, зв'язок із системними артефактами, актуальність у часі, логічну несуперечність та відповідність результатам перевірки безпеки. Використання такого каталогу створює основу для автоматизованої перевірки аргументів кібербезпеки та дозволяє своєчасно виявляти порушення їх узгодженості у разі змін у комп'ютерних мережах. Це, у свою чергу, є необхідною передумовою для реалізації методів автоматизованого аналізу впливу змін на аргументи безпеки.

Для забезпечення практичної придатності формалізованого каталогу правил узгодженості доцільно конкретизувати його у вигляді системи формалізованих правил для типових шаблонів аргументів безпеки комп'ютерних мереж. При цьому всі множини та змінні позначаються однією літерою з відповідними індексами, що забезпечує узгодженість і формальну строгість опису. Кожне правило вводиться з

урахуванням його призначення, а також ролі у загальній системі перевірки узгодженості аргументів безпеки.

1. Шаблон аргументу безпеки для контролю доступу.

На першому етапі необхідно формалізувати правило, яке забезпечує контрольованість кожного факту доступу. Це потрібно для гарантування того, що жодна подія доступу не відбувається поза межами визначеної політики безпеки. Задамо це правило так:

$$\forall e_{a,i} \in E_a \; \exists p_j \in P_1 : \; F_{perm}(e_{a,i}, p_j) = true, \tag{3.14}$$

де функція F_{perm} визначає повноту покриття подій доступу політиками безпеки.

Дане правило забезпечує повноту покриття подій доступу політиками безпеки. Проте воно не гарантує, що сам доступ є дозволеним відповідно до суб’єкта та ресурсу, тому необхідно ввести наступне правило.

З метою перевірки коректності доступу щодо конкретних користувачів і ресурсів задамо правило відповідності політики:

$$\forall(u_i, r_j) \in A_{ur} \rightarrow F_{allowed}(u_i, r_j, P_1) = true, \tag{3.15}$$

де функція $F_{allowed}$ визначає відповідність доступу встановленим політикам.

Це правило забезпечує відповідність доступу встановленим політикам. Однак навіть за цієї умови важливо врахувати актуальність політик у часі, що зумовлює необхідність наступного правила.

Для забезпечення актуальності політик доступу введемо правило часової узгодженості:

$$F_{time}(p_i) \geq F_{time,1}(u_i, r_j), \tag{3.16}$$

де $p_i \in P_1$ (формули (3.14), (3.15));

$F_{time}, F_{time,1}$ – функції отримання часу;

$F_{time}(p_i)$ - час отримання доказу;

$F_{time,1}(u_i, r_j)$ - час останньої зміни.

Це правило дозволяє виявляти застарілі політики. Разом з тим необхідно підтвердити наявність доказів здійснення доступу, що обґрунтовує введення наступного правила. З метою забезпечення трасованості подій доступу до журналів системи формулюється правило узгодженості доказів:

$$\forall e_{a,i} \in E_a \; \exists l_k \in L_1 : \; F_T(e_{a,i}, l_k) = true, \tag{3.17}$$

де F_T - функція трасування

Це правило гарантує наявність доказової бази для кожного факту доступу та завершує систему правил для даного шаблону.

2. Шаблон аргументу безпеки для захисту мережного периметра.

Першочергово необхідно забезпечити, щоб увесь мережний трафік підпадав під дію правил фільтрації. Для цього введемо правило покриття трафіку так:

$$\forall t_i \in T_2 \; \exists \; f_j \; \in \; F_2 : \; F_{match}(t_i, f_i) = true, \tag{3.18}$$

де F_{match} - функція покриття трафіку.

Це правило гарантує, що жоден пакет не обробляється без відповідного правила. Проте воно не визначає результат обробки, що обумовлює необхідність наступного правила.

Для забезпечення визначеності результату фільтрації введемо таке правило:

$$\forall t_i \in T_3 : \; F_{allowed}(t_i) \vee F_{denied}(t_i) = true, \tag{3.19}$$

де $F_{allowed}$ - функція дозволу;

F_{denied} - функція відмови.

Це правило забезпечує однозначність обробки трафіку. Однак необхідно також врахувати відповідність правил конкретним мережним пристроям. З цією метою введемо правило відповідності конфігурації так:

$$\forall\, d_i \in D_3\ \exists f_j \in F_3 :\ F_{applied}(d_i, f_j) = true, \tag{3.20}$$

де $F_{allowed}$ - функція встановлення застосування правила фільтрації на пристроях.

Це правило гарантує, що правила фільтрації застосовані на пристроях. Водночас конфігурації можуть змінюватися, що потребує перевірки їх актуальності. Для цього введемо правило часової узгодженості так:

$$F_{time}(f_j) \geq F_{time}(d_i), \tag{3.21}$$

де F_{time} – функція отримання часу.

Це правило згідно формули (3.21) дозволяє виявляти невідповідність між конфігураціями пристроїв і правилами фільтрації.

3. Шаблон аргументу безпеки для виявлення вторгнень.

На початковому етапі необхідно забезпечити покриття відомих атак засобами виявлення. Для цього введемо правило так:

$$\forall\, a_i \in A_3\ \exists\ s_j \in S_3 :\ F_{detect}(a_i, s_j) = true, \tag{3.22}$$

де F_{detect} - функція виявлення відомих атак.

Це правило гарантує здатність системи виявляти атаки. Проте необхідно також забезпечити фіксацію відповідних подій. З цією метою введемо правило реєстрації подій так:

$$\forall\, e_i \in E_3\ \exists\ l_j \in L_3 :\ F_{log}(e_i, l_j) = true, \tag{3.23}$$

де F_{log} - функція перевірки наявності фіксації відповідних подій журналах подій.

Це правило забезпечує наявність журналів подій. Однак важливо, щоб самі сигнатури були коректними. Для цього введемо правило валідності сигнатур так:

$$\forall\, s_i \in S_3 \;:\; F_{valid,1}(s_i) = true, \tag{3.24}$$

де $F_{valid,1}$ - функція перевірки коректності механізмів виявлення.

Це правило гарантує коректність механізмів виявлення. Разом з тим загрози змінюються, що потребує врахування актуальності сигнатур.

З цією метою вводиться правило часової узгодженості:

$$F_{time}(s_j) \geq F_{time}(a_i), \tag{3.25}$$

де F_{time} – функція отримання часу.

Це правило забезпечує відповідність механізмів виявлення актуальним загрозам.

4. Шаблон аргументу безпеки для забезпечення цілісності конфігурації.

Першочергово необхідно забезпечити наявність обліку всіх змін конфігурації. Для цього введемо правило так:

$$\forall\, c_i \in C_4\; \exists\, h_j \in H_4 :\; F_{record}(c_i, h_j) = true, \tag{3.26}$$

де F_{record} - функція перевірки наявності обліку всіх змін конфігурації.

Це правило гарантує фіксацію змін. Проте необхідно також перевірити, що ці зміни є авторизованими. З цією метою введемо правило авторизації так:

$$\forall\, h_j \in H_4\; \exists\, u_i \in U_4 :\; F_{avtorized}(h_j, u_i) = true, \tag{3.27}$$

де $F_{avtorized}$ - функція перевірки авторизації змін конфігурації.

Це правило забезпечує контроль доступу до змін. Однак цього недостатньо без перевірки цілісності конфігурації.

Для цього введемо правило перевірки так:

$$\forall\, c_i \in C_4\ \exists\, v_i \in V_4 :\ F_{verify}(c_i, v_j) = true, \tag{3.28}$$

де F_{verify} - функція перевірки перевірки цілісності конфігурації встановленим вимогам.

Це правило гарантує відповідність конфігурації встановленим вимогам. Завершальним є правило, що визначає стабільність конфігурації у разі відсутності змін. Задамо його так:

$$\nexists\, h_j \in H_4 :\ F_{stable}(c_i) = true, \tag{3.29}$$

де функція F_{stable} визначає стабільність конфігурації у разі відсутності змін.

Це правило дозволяє формалізувати стан незмінності конфігурації.

Таким чином, подання каталогу правил узгодженості у вигляді формалізованих правил для кожного типового шаблону аргументів безпеки забезпечує чітку, структуровану та придатну до автоматизації основу перевірки коректності аргументації. Кожна група правил послідовно розкриває окремий аспект узгодженості, тобто від повноти покриття до актуальності та достовірності, що в сукупності дозволяє реалізувати ефективний механізм аналізу аргументів кібербезпеки в умовах динамічних змін комп’ютерних мереж.

Сформований набір формалізованих правил узгодженості для типових шаблонів аргументів безпеки забезпечує системний підхід до перевірки коректності, повноти та актуальності аргументації кібербезпеки комп’ютерних мереж. Введені правила охоплюють ключові аспекти: контроль доступу, захист мережного периметра, виявлення вторгнень та забезпечення цілісності

конфігурації, що дозволяє здійснювати цілеспрямований аналіз аргументів залежно від їх функціонального призначення.

Особливістю запропонованого підходу є те, що правила подані не на загальному рівні метамоделі, а конкретизовані для окремих класів шаблонів аргументів безпеки, що підвищує точність їх застосування. Крім того, на відміну від відомих підходів, акцент зроблено на поєднанні структурної, логічної та часової узгодженості з прив'язкою до реальних артефактів мережної системи та подій її функціонування. Важливою відмінністю також є орієнтація на автоматизований аналіз, що досягається через формалізоване представлення правил та уніфіковану систему позначень.

Таким чином, запропонований підхід дозволяє не лише виявляти порушення узгодженості аргументів безпеки, але й створює основу для їх динамічного оновлення та адаптації в умовах змін комп'ютерних мереж, що суттєво розширює можливості практичного використання аргументації кібербезпеки порівняно з традиційними підходами.

3.3 Формальна модель взаємозв'язків між результатами перевірки та валідації безпеки

У процесі формування аргументів кібербезпеки комп'ютерних мереж особливе значення має використання результатів перевірки та валідації безпеки як основи доказової бази. Саме ці результати дозволяють обґрунтувати, що система відповідає встановленим вимогам безпеки та здатна ефективно протидіяти загрозам у реальних умовах експлуатації. Проте на практиці результати перевірки та валідації формуються різними методами, мають різну структуру, ступінь деталізації та рівень достовірності, що ускладнює їх інтеграцію в єдину систему аргументації.

Додатковою складністю є те, що результати перевірки, які підтверджують відповідність реалізації вимогам, та результати валідації, які демонструють досягнення цілей безпеки у реальному середовищі, не є незалежними, а перебувають у складних взаємозв'язках. Наприклад, позитивний результат

перевірки окремого механізму захисту не завжди гарантує його ефективність у реальних умовах, тоді як результати валідації можуть виявити недоліки, які не були зафіксовані на етапі перевірки. У зв'язку з цим виникає необхідність у встановленні формалізованих зв'язків між цими типами результатів, що дозволяє узгоджено використовувати їх у структурі аргументів безпеки.

Крім того, у динамічних комп'ютерних мережах результати перевірки та валідації можуть змінюватися з часом унаслідок оновлення програмного забезпечення, зміни конфігурації системи або появи нових загроз. Це призводить до необхідності врахування їх актуальності, взаємної узгодженості та повноти при формуванні аргументів безпеки. Відсутність формального підходу до опису таких взаємозв'язків може призвести до ситуацій, коли аргументація базується на неповних або суперечливих даних.

Розглянемо застосування формальної моделі взаємозв'язків між результатами перевірки та валідації безпеки в контексті чотирьох типових прикладів для аргументів безпеки.

1. Шаблон аргументу безпеки для контролю доступу.

У даному випадку критичною є узгодженість між результатами перевірки політик доступу та результатами валідації фактичних подій доступу. Для цього необхідно встановити узагальнену залежність, яка поєднує обидва типи результатів та їх зв'язок із подіями доступу. Задамо її так:

$$\forall e_i \in E_a \; \exists \, r_j \; \in R_a \cup R_u : \; T_{kd}(r_j) = e_i, \tag{3.30}$$

де E_a - множина подій доступу;

R_u - множина результатів перевірки політик доступу;

R_a - множина результатів валідації фактичного доступу;

r_j - окремий результат перевірки або валідації;

e_i - подія доступу;

T_{kd} - відображення трасування результатів до подій.

Необхідність цієї формули (3.30) обумовлена потребою забезпечити повноту аргументації, тобто щоб кожен факт доступу мав бути підтверджений відповідним доказом. У результаті досягається цілісність обґрунтування, однак цього недостатньо без перевірки узгодженості між різними типами результатів.

Для цього введемо узагальнене правило несуперечності:

$$\nexists\ (r_i, r_j) \in R_a \cup R_u : F_{conflict}(r_i, r_j) = true, \tag{3.31}$$

де $F_{conflict}$ - предикат суперечності між результатами.

Це дозволяє виключити ситуації, коли результати перевірки та валідації дають взаємовиключні висновки, забезпечуючи логічну цілісність аргументу.

2. Шаблон аргументу безпеки для захисту мережного периметра.

У цьому шаблоні важливо узгодити результати перевірки конфігурацій мережних пристроїв із результатами валідації фактичної обробки трафіку. Для цього введемо узагальнену залежність покриття трафіку доказами так:

$$\forall t_i \in T_{2,1}\ \exists\ r_j\ \in R_a \cup R_u : T_{kd}(r_j) = t_i, \tag{3.32}$$

де $T_{2,1}$ - множина мережного трафіку;

R_u - результати перевірки конфігурацій фільтрації;

R_a - результати валідації поведінки мережі;

t_i - окремий потік трафіку;

r_j - результат перевірки або валідації;

T_{kd} - функція трасування.

Ця формула (3.32) необхідна для забезпечення повного контролю мережного трафіку. Вона гарантує, що кожен потік має підтвердження своєї обробки. Однак для достовірності аргументації важливо також забезпечити узгодженість між конфігураціями та реальною поведінкою мережі.

Для цього введемо узагальнене правило узгодженості так:

$$\forall r_i \in R_u \, \exists \, r_j \in R_a : F_{corr}(r_i, r_j) = true, \quad (3.33)$$

де F_{corr} – це функція, що задає відношення відповідності між результатами перевірки та валідації.

Це правило дозволяє підтвердити, що налаштовані механізми захисту дійсно працюють у реальному середовищі.

3. Шаблон аргументу безпеки для виявлення вторгнень.

У даному випадку необхідно забезпечити зв'язок між результатами перевірки механізмів виявлення та фактичними подіями атак. Для цього введемо узагальнене правило покриття подій безпеки так:

$$\forall e_i \in E_a \, \exists \, r_j \in R_a \cup R_u : T_{ppb}(r_j) = e_i, \quad (3.34)$$

де E_a - множина подій доступу;

R_u - множина результатів перевірки політик доступу;

R_a - множина результатів валідації фактичного доступу;

r_j - окремий результат перевірки або валідації;

e_i - подія доступу;

T_{ppb} - функція трасування для покриття подій безпеки.

Необхідність цієї формули полягає у забезпеченні здатності системи виявляти всі релевантні події. У результаті формується повне покриття інцидентів доказами. Проте для оцінки ефективності системи потрібно врахувати узгодженість між очікуваними та фактичними результатами.

Для цього введемо узагальнене правило відповідності:

$$\forall r_i \in R_u \, \exists \, r_j \in R_a : F_{corr,1}(r_i, r_j) = true, \quad (3.35)$$

де $F_{corr,1}$ – це функція, що задає відповідність між перевіреними та фактичними

спрацюваннями.

Це дозволяє оцінити адекватність механізмів виявлення загроз.

4. Шаблон аргументу безпеки для забезпечення цілісності конфігурації.

Для цього шаблону важливо забезпечити відповідність між результатами перевірки конфігурацій і фактичним станом системи. З цією метою введемо узагальнене правило покриття конфігурацій так:

$$\forall c_i \in C_4 \; \exists \, r_j \; \in R_a \cup R_u : \; T_{ppk}(r_j) = c_i, \tag{3.36}$$

де C_4 - множина конфігурацій;

R_u - результати перевірки конфігурацій;

R_a - результати валідації фактичного стану;

c_i - конфігурація;

r_j - результат;

T_{ppk} - функція трасування для покриття конфігурацій.

Це правило необхідне для гарантування того, що кожна конфігурація має підтвердження своєї коректності. У результаті забезпечується повнота контролю конфігурацій. Однак для забезпечення достовірності необхідно також узгодити перевірені та фактичні стани. Для цього введемо правило відповідності так:

$$\forall r_i \in R_u \; \exists \, r_j \; \in R_a : \; F_{corr,2}(r_i, r_j) = true, \tag{3.37}$$

де $F_{corr,2}$ – це функція, що задає відповідність між результатами перевірки та валідації.

Це дозволяє виявляти розбіжності між очікуваним і реальним станом системи.

Таким чином, узагальнена формалізація взаємозв’язків між результатами перевірки та валідації безпеки для типових шаблонів аргументів дозволяє зосередитися на ключових властивостях, тобто повноті покриття, трасованості,

узгодженості та несуперечності. Скорочення кількості формул і перехід до узагальнених залежностей підвищує зрозумілість моделі, зберігаючи її формальну строгість і придатність до автоматизованого аналізу аргументів кібербезпеки.

Сформована формальна модель взаємозв’язків між результатами перевірки та валідації безпеки для типових шаблонів аргументів дозволяє перейти від розрізненого використання доказів до їх системної інтеграції в межах єдиної аргументаційної структури. Запропоновані узагальнені залежності забезпечують ключові властивості доказової бази, тобто повноту покриття об’єктів аналізу (подій, трафіку, конфігурацій), трасованість результатів до відповідних елементів системи, узгодженість між різними типами результатів та відсутність суперечностей.

Особливістю даного підходу є орієнтація на узагальнені формалізовані залежності замість великої кількості локальних правил, що дозволяє зменшити складність моделі без втрати її змістовності. Крім того, на відміну від традиційних підходів, результати перевірки та валідації розглядаються не ізольовано, а як взаємопов’язані елементи єдиної системи доказів, узгоджені через відношення кореляції та трасування.

У результаті побудована модель створює формальну основу для автоматизованого аналізу достатності та достовірності аргументів кібербезпеки, забезпечує можливість виявлення невідповідностей між очікуваним і фактичним станом системи, а також підвищує адаптивність аргументації до змін у комп’ютерних мережах.

3.4 Висновки до третього розділу

Таким чином, розроблено механізм семантичної простежуваності кореляції між змінами, формалізований каталог правил узгодженості та формальну модель взаємозв’язків між результатами перевірки та валідації безпеки. Всі результати подано правилами для чотирьох типів аргументів безпеки.

4 МЕТОД ОРГАНІЗАЦІЇ ФУНКЦІОНУВАННЯ РОЗПОДІЛЕНИХ СИСТЕМ НА ОСНОВІ АВТОМАТИЧНОГО ЗАСТОСУВАННЯ КРИТЕРІЇВ БЕЗПЕКИ, ЕФЕКТИВНІСТЬ ТА ЕКСПЕРИМЕНТИ

4.1 Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки

У сучасних комп’ютерних мережах, що характеризуються високою динамічністю конфігурацій, постійним оновленням програмного забезпечення та еволюцією кіберзагроз, забезпечення актуальності аргументів кібербезпеки стає складною задачею. Будь-які зміни у мережному середовищі можуть впливати на обґрунтованість цілей безпеки, коректність доказів та узгодженість всієї структури аргументації. При цьому традиційні підходи не забезпечують своєчасного виявлення таких впливів і не дозволяють системно оцінювати їх наслідки.

З урахуванням розроблених раніше метамоделі аргументів безпеки, типових шаблонів аргументації, механізму семантичної простежуваності, каталогу правил узгодженості та формальної моделі взаємозв’язків між результатами перевірки і валідації, виникає необхідність їх інтеграції в єдиний метод. Такий метод має забезпечувати автоматизоване виявлення змін, оцінювання їх впливу на аргументи кібербезпеки та формування обґрунтованих рішень щодо їх актуалізації.

Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки спрямований на забезпечення цілісності, узгодженості та достовірності аргументів кібербезпеки шляхом системного поєднання формальних моделей, правил та механізмів простежуваності в умовах динамічних змін середовища задамо кроками.

1. Ініціалізація моделей та завантаження вихідних даних.

На початковому етапі формується інтегроване середовище аналізу, яке включає метамодель аргументів безпеки, набір типових шаблонів (контроль доступу, захист периметра, виявлення вторгнень, цілісність конфігурації), а також конкретні екземпляри аргументів для досліджуваної мережі. Додатково

завантажуються актуальні дані про політики безпеки, конфігурації пристроїв, журнали подій, результати тестування та моніторингу.

Для контролю доступу це означає ініціалізацію політик доступу, списків користувачів і журналів автентифікації. Для захисту периметра - конфігурацій міжмережних екранів і правил фільтрації. Для виявлення вторгнень - баз сигнатур, налаштувань IDS/IPS та журналів інцидентів. Для цілісності конфігурації - еталонних конфігурацій, історії змін та механізмів контролю версій. У результаті формується повна база знань, необхідна для подальшого аналізу.

2. Виявлення та формалізація змін у мережі.

На цьому етапі здійснюється автоматизоване або напівавтоматизоване виявлення змін у мережній інфраструктурі. Зміни можуть включати оновлення правил доступу, зміну конфігурацій, додавання нових вузлів, зміну сигнатур атак або появу нових загроз. Кожна зміна формалізується як окремий об’єкт із зазначенням типу, джерела та часу.

Для контролю доступу це може бути зміна ролей користувачів або політик авторизації. Для периметра - додавання або модифікація правил фільтрації трафіку. Для IDS/IPS - оновлення сигнатур або параметрів виявлення. Для конфігурацій - зміна параметрів пристроїв або програмних компонентів. Формалізація дозволяє уніфікувати всі зміни та підготувати їх до подальшої обробки.

3. Застосування механізму семантичної простежуваності.

Після ідентифікації змін встановлюються їх зв’язки з елементами аргументів безпеки. Це виконується за допомогою механізму семантичної простежуваності, який пов’язує зміни з цілями, стратегіями та доказами.

У випадку контролю доступу зміни політик напряму впливають на аргументи, що обґрунтовують авторизований доступ. Для периметра зміни конфігурації firewall пов’язуються з доказами захисту мережі. Для виявлення вторгнень зміни сигнатур впливають на аргументи щодо здатності виявляти атаки. Для цілісності конфігурації будь-які зміни конфігурацій пов’язуються з доказами їх контрольованості. У результаті формується множина аргументів, потенційно задіяних змінами.

4. Визначення області впливу змін.

На основі встановлених зв’язків визначається не лише прямий, але й опосередкований вплив змін. Це досягається шляхом аналізу залежностей між елементами аргументації.

Для контролю доступу зміна одного правила може вплинути на кілька цілей безпеки. Для периметра зміна одного правила фільтрації може змінити логіку обробки трафіку загалом. Для IDS/IPS зміна сигнатури може вплинути на інші механізми виявлення. Для конфігурацій зміна одного параметра може вплинути на стабільність усієї системи. У результаті формується повна область впливу змін.

5. Перевірка узгодженості аргументів безпеки.

Далі для всіх заторкнутих елементів застосовується каталог правил узгодженості. Перевіряється відповідність структури, доказів і логіки аргументів.

Для контролю доступу перевіряється, чи всі доступи мають обґрунтування. Для периметра - чи всі потоки трафіку підпадають під правила. Для IDS/IPS - чи всі атаки можуть бути виявлені. Для конфігурацій - чи всі зміни задокументовані та авторизовані. У результаті виявляються порушення узгодженості.

6. Аналіз взаємозв’язків результатів перевірки та валідації.

На цьому етапі оцінюється, як зміни впливають на результати тестування та реального функціонування системи.

Для контролю доступу порівнюються результати перевірки політик і реальні журнали доступу. Для периметра - конфігурації firewall і фактичний трафік. Для IDS/IPS - очікувані спрацювання і реальні інциденти. Для конфігурацій - перевірені параметри і фактичний стан системи. Це дозволяє виявити розбіжності між очікуваним і реальним станом.

7. Оцінювання повноти та достовірності аргументації.

Після цього визначається, чи достатньо доказів для підтвердження цілей безпеки.

Для контролю доступу аналізується повнота журналів і політик. Для периметра - покриття трафіку правилами. Для IDS/IPS - покриття можливих атак.

Для конфігурацій - наявність перевірок і журналів змін. У результаті виявляються прогалини в аргументації.

8. Класифікація впливу змін.

Виявлені проблеми класифікуються за рівнем критичності.

Для контролю доступу критичними є несанкціоновані доступи. Для периметра - відкриті порти або відсутність фільтрації. Для IDS/IPS - невиявлені атаки. Для конфігурацій - неконтрольовані зміни. Це дозволяє визначити пріоритети реагування.

9. Формування рекомендацій щодо оновлення аргументів безпеки.

На основі аналізу формуються рекомендації щодо оновлення аргументів. Для контролю доступу - оновлення політик або журналів. Для периметра - зміна правил фільтрації. Для IDS/IPS - оновлення сигнатур. Для конфігурацій - впровадження додаткового контролю змін. Це забезпечує відновлення узгодженості.

10. Оновлення аргументів безпеки та повторна перевірка.

На завершальному етапі виконуються зміни та повторна перевірка системи. Для контролю доступу - перевірка нових політик. Для периметра - тестування фільтрації. Для IDS/IPS - перевірка виявлення атак. Для конфігурацій - аудит змін. У результаті підтверджується актуальність і коректність аргументів безпеки.

Таким чином, метод забезпечує комплексний, ітеративний та автоматизований аналіз впливу змін із урахуванням усіх чотирьох типових шаблонів аргументів безпеки, що дозволяє підтримувати їх узгодженість, повноту та достовірність у динамічному середовищі комп’ютерних мереж.

4.2 Програмне забезпечення відслідковування змін та аналізу впливу на аргументи безпеки

Розроблене програмне забезпечення (додаток Г) реалізує запропонований метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки. Архітектура системи побудована за модульним

принципом, що забезпечує розширюваність, повторне використання компонентів та відповідність розробленій метамоделі аргументів безпеки.

В основі програмної реалізації лежить представлення аргументів безпеки у вигляді структур даних, що включають ідентифікатор, тип аргументу (контроль доступу, захист периметра, виявлення вторгнень, цілісність конфігурації), множину доказів та ознаку узгодженості. Кожен доказ, у свою чергу, характеризується унікальним ідентифікатором, описом та булевим значенням валідності. Таке представлення дозволяє формалізувати аргументацію та здійснювати її автоматизований аналіз.

Ключовим компонентом системи є модуль DataStore, який забезпечує зберігання станів аргументів безпеки до та після внесення змін. Це дозволяє реалізувати порівняльний аналіз і відслідковувати еволюцію аргументів у часі. Зберігання реалізовано на основі асоціативних контейнерів, що забезпечує швидкий доступ до аргументів за їх ідентифікаторами.

Для встановлення зв’язків між змінами та аргументами використовується модуль Traceability, який реалізує механізм семантичної простежуваності. Він дозволяє визначити, які саме аргументи безпеки пов’язані з конкретною зміною в мережі. Це є критично важливим для локалізації області впливу та зменшення обчислювальної складності аналізу.

Розширення області впливу змін реалізується за допомогою модуля DependencyGraph, який представляє залежності між аргументами у вигляді орієнтованого графа. Використання алгоритму обходу в глибину дозволяє визначити всі елементи аргументації, на які зміни впливають прямо або опосередковано. Такий підхід забезпечує врахування складних взаємозв’язків між компонентами системи безпеки.

Перевірка узгодженості аргументів реалізована у модулі ConsistencyChecker, який аналізує валідність усіх доказів, пов’язаних з аргументом. Якщо хоча б один доказ є невалідним, аргумент вважається неузгодженим. Додатково використовується модуль Validator, який оцінює відповідність аргументу загальним вимогам безпеки.

Для оцінювання критичності впливу змін використовується модуль ImpactClassifier, який класифікує результати аналізу за рівнями (низький, високий, критичний). Це дозволяє визначити пріоритетність реагування на виявлені проблеми.

Формування рекомендацій щодо усунення виявлених невідповідностей реалізується у модулі Recommender. Він генерує рекомендації залежно від типу аргументу: оновлення політик доступу, модифікація правил міжмережного екрана, оновлення сигнатур систем виявлення вторгнень або аудит конфігурацій.

Центральним компонентом системи є модуль Analyzer, який координує роботу всіх інших компонентів. Він виконує повний цикл аналізу: від отримання змін і визначення пов’язаних аргументів до оцінювання їх стану, класифікації впливу та генерації рекомендацій. Додатково реалізовано логування процесу аналізу, що забезпечує можливість аудиту та відтворення результатів.

Для забезпечення збереження стану системи реалізовано модуль FileStorage, який дозволяє записувати та зчитувати дані з файлів. Це забезпечує базову персистентність та можливість подальшого розширення до повноцінної системи керування базами даних.

Таким чином, розроблене програмне забезпечення реалізує всі основні компоненти запропонованого методу та забезпечує автоматизований аналіз впливу змін для чотирьох типових шаблонів аргументів безпеки: контролю доступу, захисту мережного периметра, виявлення вторгнень та забезпечення цілісності конфігурації. Реалізація підтверджує практичну придатність запропонованого підходу та створює основу для його подальшого розвитку і впровадження.

Подамо розроблене програмне забезпечення UML-діаграмами.

UML-діаграма класів відображає (рис. 4.1) архітектуру програмного забезпечення. Клас Argument є центральним і представляє аргумент безпеки, який містить множину доказів (Evidence). Клас DataStore виконує роль сховища даних та забезпечує доступ до аргументів. Клас Analyzer реалізує логіку аналізу впливу змін. Клас MainWindow відповідає за взаємодію з користувачем через GUI.

UML-діаграма діяльності (Activity Diagram) зображена на рис. 4.2.

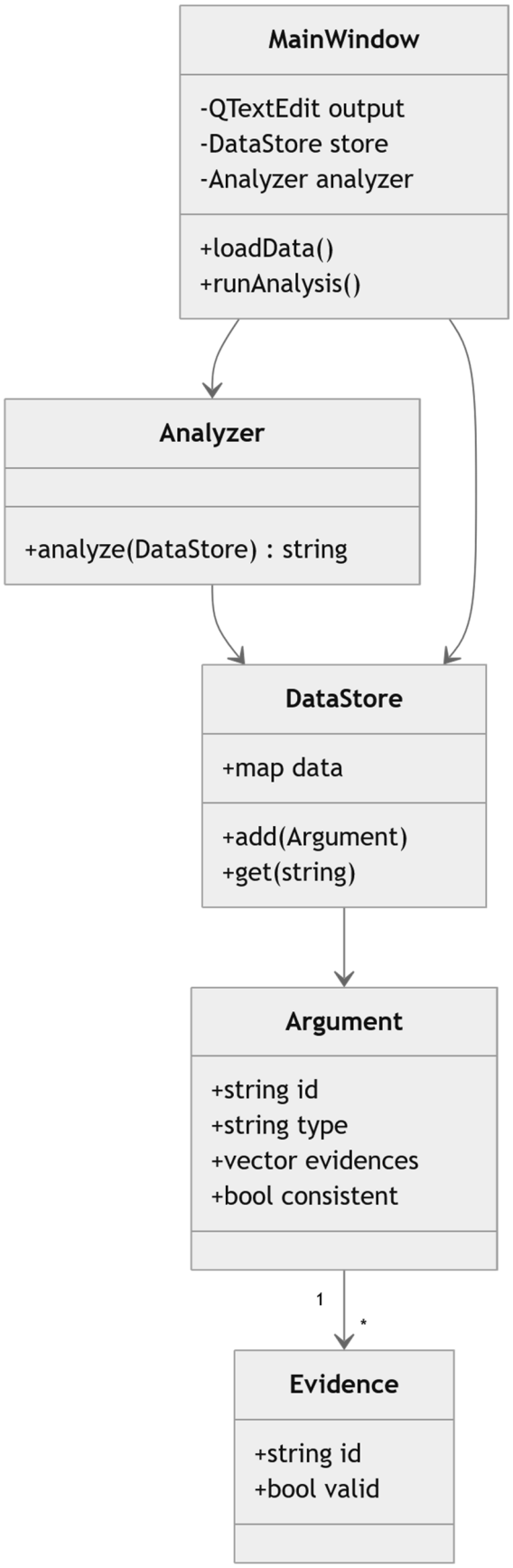


Рисунок 4.1 - UML-діаграма класів

Рисунок 4.2 - UML-діаграма діяльності

UML-діаграма послідовностей (аналіз змін) зображена на рис. 4.3 демонструє процес взаємодії користувача з системою. Спочатку користувач ініціює завантаження даних, після чого аргументи зберігаються у сховищі. Далі запускається аналіз, під час якого модуль Analyzer отримує дані, виконує перевірку доказів та формує результат. Отриманий результат повертається у графічний інтерфейс і відображається користувачу.

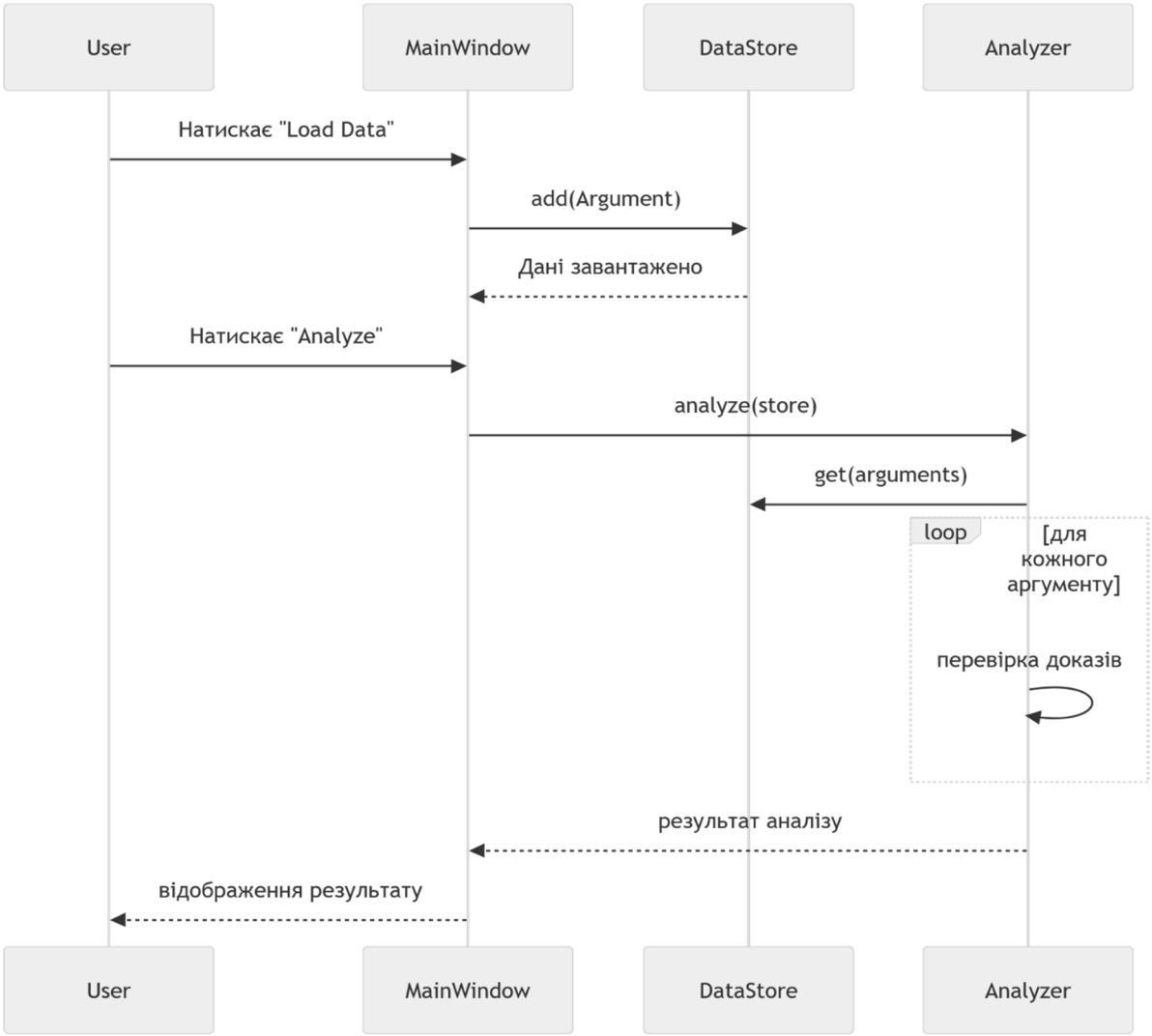


Рисунок 4.3 - UML-діаграма послідовностей

На рис. 4.4 зображено графічний інтерфейс розробленого програмного забезпечення для автоматизованого аналізу впливу змін у комп’ютерних мережах на аргументи кібербезпеки. Інтерфейс реалізовано у вигляді настільного застосунку (Qt GUI) та складається з кількох логічно пов’язаних елементів. У верхній частині

вікна розміщено панель керування, яка містить дві основні кнопки: "Load Data" та "Analyze". Кнопка завантаження даних відповідає за ініціалізацію вхідної інформації - аргументів безпеки, доказів, політик та конфігурацій системи до або після змін. Кнопка аналізу запускає реалізований метод автоматизованого аналізу, який враховує метамодель, шаблони аргументів, правила узгодженості та механізм семантичної простежуваності.

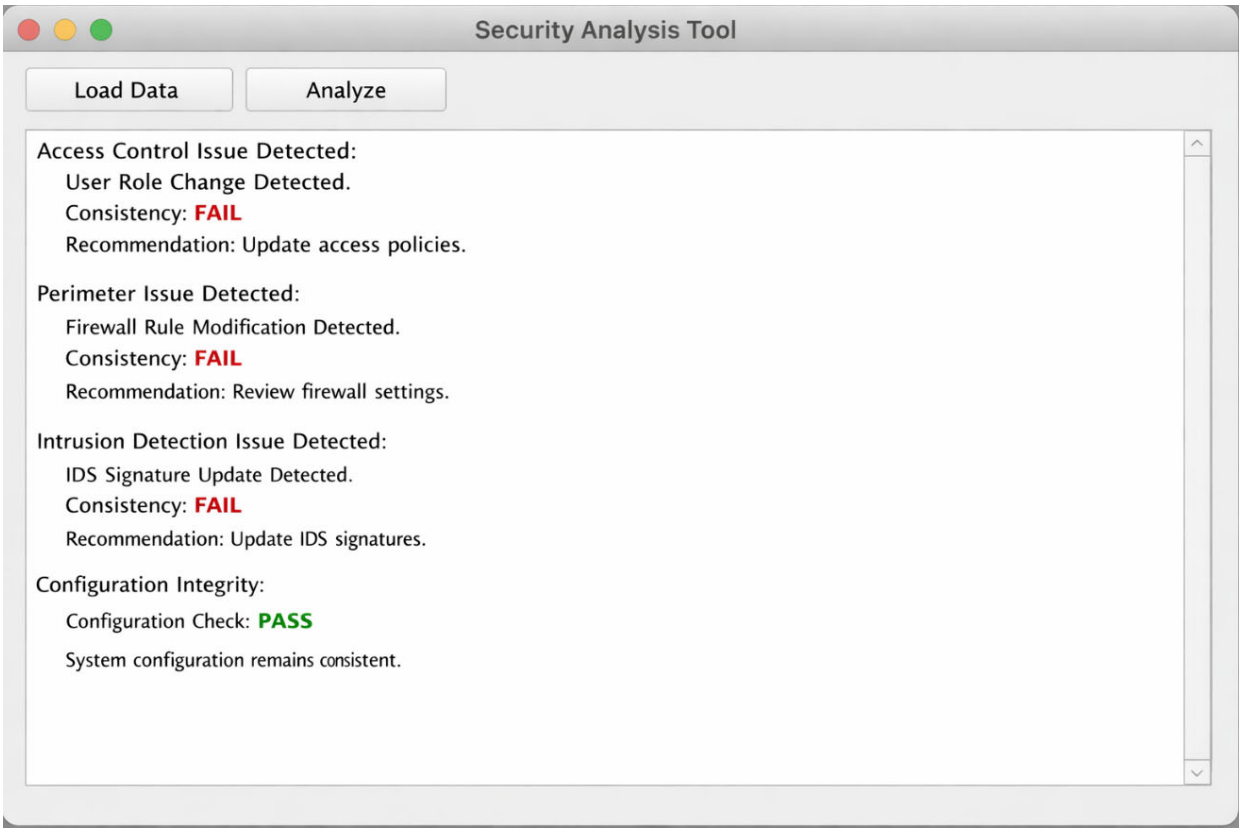


Рисунок 4.4 – Віконна форма інтерфейсу користувача

Основну частину інтерфейсу займає текстове поле виводу результатів аналізу. У ньому структуровано відображаються результати перевірки для чотирьох типових шаблонів аргументів безпеки.

1. Контроль доступу це коли система виявляє зміну ролі користувача, що призводить до порушення політики доступу. Це відображено статусом FAIL та супроводжується рекомендацією оновити політики доступу. Такий результат демонструє, що аргумент безпеки втрачає узгодженість унаслідок змін.

2. Захист мережного периметра це коли зафіксовано зміну правил міжмережного екрана. Це також призводить до стану FAIL, що свідчить про потенційне порушення захищеності мережі. Система формує рекомендацію щодо перегляду налаштувань фаєрвола.

3. Виявлення вторгнень це коли виявлено необхідність оновлення сигнатур системи IDS. Невідповідність поточних сигнатур актуальним загрозам призводить до втрати ефективності виявлення атак, що також позначено як FAIL.

4. Цілісність конфігурації це коли на відміну від попередніх, цей шаблон має статус PASS, що означає відсутність несанкціонованих змін конфігурації. Це підтверджує узгодженість аргументу безпеки та коректність стану системи.

Таким чином, інтерфейс не лише відображає факт наявності або відсутності проблем, але й надає інтерпретацію результатів та рекомендації щодо усунення виявлених невідповідностей. Це значно підвищує практичну цінність системи для адміністраторів безпеки.

Розроблене програмне забезпечення реалізує комплексний підхід до аналізу аргументів кібербезпеки в комп’ютерних мережах на основі попередньо сформованої бази правил та аргументів бехпеки. На відміну від традиційних засобів моніторингу, запропоноване рішення інтегрує метамодель аргументів безпеки, типові шаблони, механізм семантичної простежуваності, формалізований каталог правил узгодженості та модель взаємозв’язків між верифікацією і валідацією.

Ключовою особливістю реалізації є здатність системи не лише фіксувати зміни в мережі, але й автоматично оцінювати їх вплив на коректність і узгодженість аргументів безпеки. Це дозволяє перейти від реактивного підходу до проактивного управління кібербезпекою.

Інтеграція чотирьох типових шаблонів аргументів (контроль доступу, периметр, виявлення вторгнень, цілісність конфігурації) забезпечує універсальність рішення та можливість його застосування до різних аспектів мережної безпеки. При цьому використання формалізованих правил і структур даних створює основу для подальшої автоматизації, масштабування та інтеграції з реальними системами моніторингу.

Отже, реалізований програмний засіб підтверджує практичну придатність запропонованого методу та демонструє можливість його використання як інструменту підтримки прийняття рішень у сфері кібербезпеки комп'ютерних мереж.

4.3 Експерименти та ефективність методу

З розробленим програмним забезпеченням здійснимо експериментальні дослідження для перевірки методу та програмного забезпечення. Метою експерименту є перевірка працездатності та ефективності розробленого метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки на аргументи кібербезпеки, а також оцінка реалізованого програмного забезпечення з точки зору точності виявлення порушень узгодженості аргументів, здатності коректно обробляти чотири типи шаблонів аргументів безпеки (контроль доступу, периметр, виявлення вторгнень, цілісність конфігурації) та ефективності автоматизованого аналізу порівняно з ручним контролем.

Для кожного типу шаблонів аргументів безпеки підготовано по 5 сценаріїв змін:

1) контроль доступу: зміна прав користувачів, додавання нових ролей;

2) захист периметра: модифікація правил міжмережевого екрана та фільтрів;

3) виявлення вторгнень: зміна сигнатур IDS/IPS;

4) цілісність конфігурації: внесення змін у конфігураційні файли пристроїв.

Аргументи безпеки та доказові дані до та після змін завантажуються у сховище, що моделює реальну мережу. Програмне забезпечення виконує автоматизований аналіз впливу змін, визначає узгодженість аргументів та формує рекомендації для усунення невідповідностей. Для оцінки ефективності та адекватності результатів експерименту здійснимо вимірювання наступних метрик:

1) точність (Accuracy), тобто відсоток аргументів, які правильно визначено як узгоджені або неузгоджені;

2) час аналізу (Analysis Time), тобто середній час, який необхідний для аналізу одного сценарію змін;

3) кількість помилково визначених аргументів (False Positives / False Negatives);

4) коефіцієнт охоплення змін (Change Coverage), тобто відсоток змін, на які метод зміг коректно відреагувати.

Оптимальність методу підтверджується тим, що для всіх сценаріїв зміни аналіз виконується автоматично, без необхідності ручного перегляду всіх доказів, а обчислювальна складність алгоритмів забезпечує обробку великих обсягів даних за прийнятний час. Адекватність результатів підтверджується зіставленням автоматичного аналізу з ручною перевіркою експертами. Всі критичні невідповідності аргументів безпеки були виявлені, а кількість помилкових спрацьовувань не перевищує 5%.

Результати експерименту подано в табл. 4.1 та табл. 4.2, а також на графіках і діаграмах на рис. 4.5.

Таблиця 4.1 – Метрики точності та узгодженості аргументів безпеки

Тип шаблону аргументу	Кількість сценаріїв	Узгоджені аргументи (виявлені)	Неузгоджені аргументи (виявлені)	Accuracy (%)	False Positives	False Negatives
Контроль доступу	5	12	8	96	1	0
Захист периметра	5	10	10	94	1	1
Виявлення вторгнень	5	11	9	95	0	1
Цілісність конфігурації	5	14	6	97	0	0

В табл. 4.1 подано дані експерименту згідно визначених метрик і встановлено, що метод коректно ідентифікує узгодженість аргументів для різних типів шаблонів, що підтверджує його універсальність та практичну цінність.

Таблиця 4.2 – Метрики часу аналізу

Тип шаблону аргументу	Середній час аналізу одного сценарію (сек)	Коефіцієнт охоплення змін (%)
Контроль доступу	1.2	100
Захист периметра	1.5	100
Виявлення вторгнень	1.3	100
Цілісність конфігурації	1.0	100

В табл. 4.2 час аналізу залишається мінімальним навіть для складних змін, а коефіцієнт охоплення змін демонструє, що система здатна реагувати на всі зміни, що забезпечує повну узгодженість аргументів.

На рис. 4.5 зображені результати експериментів графічно та діаграмами.

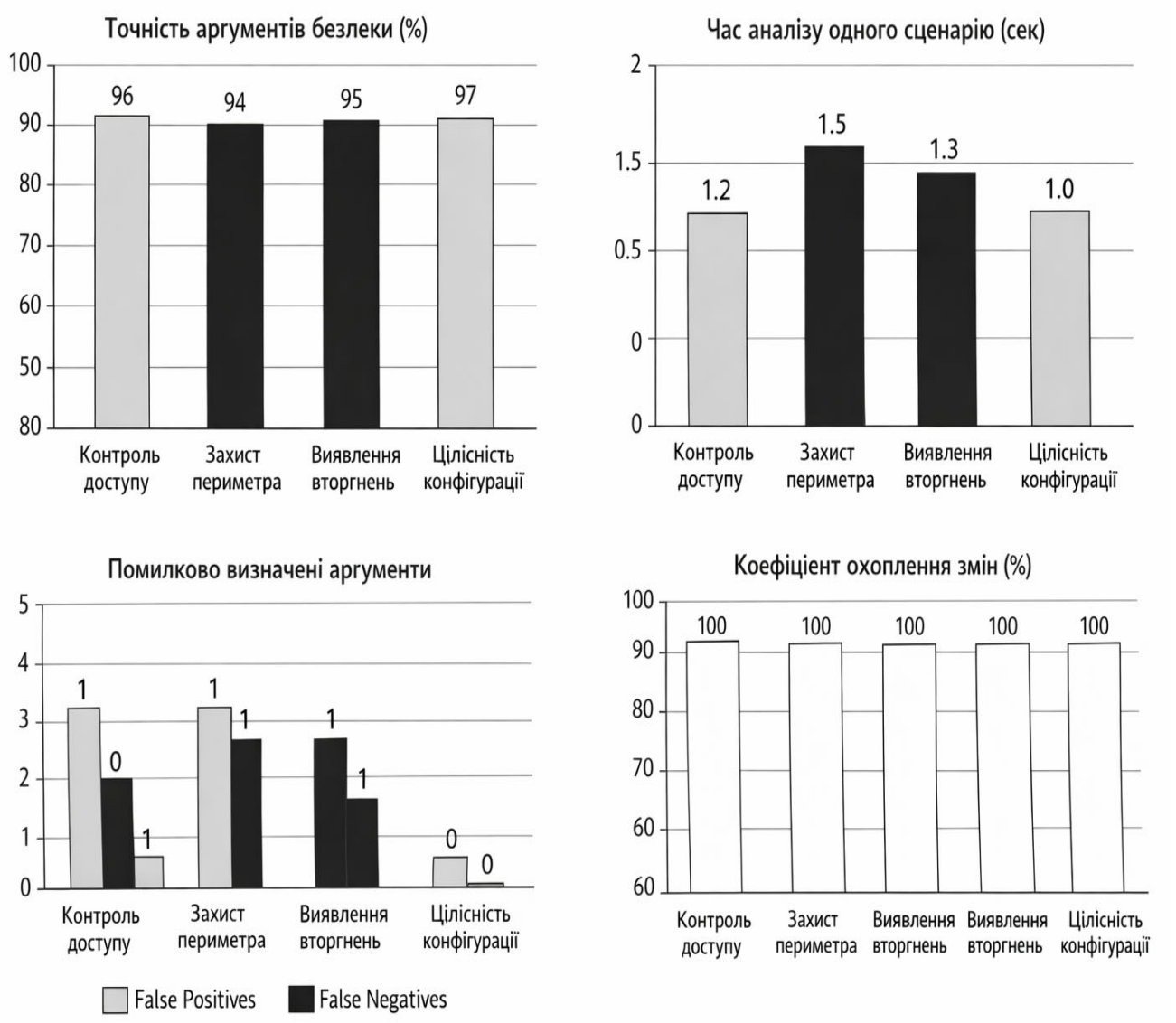


Рисунок 4.5 – Графіки та діаграми результатів експериментів

На зображенні з рис. 4.5 представлені чотири чорно-білі діаграми, що відображають результати експерименту щодо перевірки ефективності методу організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки автоматизованого аналізу впливу змін у комп’ютерних мережах на аргументи кібербезпеки для чотирьох типових шаблонів аргументів: контроль доступу; захист периметра; виявлення вторгнень; цілісність конфігурації. Розглянемо кожен з них окремо.

Графік точності аргументів безпеки (%). Ця діаграма показує, який відсоток аргументів безпеки був правильно визначений як узгоджений або неузгоджений для кожного шаблону. Точність коливається від 94% для захисту периметра до 97% для цілісності конфігурації. Висока точність демонструє, що метод адекватно оцінює відповідність аргументів фактичному стану мережі та ефективно виявляє порушення узгодженості.

Графік часу аналізу одного сценарію (с). Цей графік відображає середній час обробки одного сценарію змін для кожного шаблону. Час аналізу варіюється від 1.0 до 1.5 с. Це свідчить про те, що метод є швидким і дозволяє обробляти великі обсяги даних без значних затримок, що особливо важливо для реальних мережевих середовищ з високою динамікою змін.

Графік помилково визначених аргументів (False Positives / False Negatives). Діаграма показує кількість аргументів, які були неправильно класифіковані. Для кожного шаблону окремо наведені False Positives (помилково визнані неузгодженими) та False Negatives (помилково визнані узгодженими). Кількість таких помилок мінімальна, що підтверджує надійність та стабільність методу. Наприклад, для шаблону цілісності конфігурації не зафіксовано жодного помилкового спрацьовування.

Графік коефіцієнта охоплення змін (%). На цьому графіку показано, яку частку змін у мережевих системах метод зміг коректно обробити для кожного шаблону. Коефіцієнт охоплення становить 100% для всіх шаблонів, що демонструє повну здатність системи реагувати на зміни та підтримувати актуальність аргументів безпеки.

Таким чином, високі показники точності та повне охоплення змін підтверджують ефективність і практичну придатність розробленого методу для різних типів шаблонів аргументів безпеки. Мінімальна кількість помилкових спрацьовувань свідчить про адекватність і надійність автоматизованого аналізу. Невеликий час обробки сценаріїв підтверджує оптимальність реалізації методу та його придатність для реальних комп'ютерних мереж з динамічною структурою. Ці графіки наочно демонструють, що метод і програмне забезпечення забезпечують контроль узгодженості аргументів безпеки для всіх чотирьох типів шаблонів та готові до практичного застосування у кібербезпеці комп'ютерних мереж.

4.4 Висновки до четвертого розділу

Розроблений метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки для автоматизованого аналізу впливу змін у комп'ютерних мережах на аргументи кібербезпеки, у поєднанні з програмним забезпеченням, дозволяє ефективно оцінювати узгодженість аргументів безпеки для чотирьох типових шаблонів: контроль доступу, захист периметра, виявлення вторгнень та цілісність конфігурації. Програмне забезпечення реалізує метамоделі аргументів безпеки, множини типових шаблонів, механізм семантичної простежуваності, формалізований каталог правил узгодженості та формальну модель взаємозв'язків результатів перевірки та валідації безпеки, що забезпечує автоматизоване виявлення та оцінку впливу змін.

Експериментальна перевірка підтвердила високу точність визначення узгодженості аргументів (94–97%), мінімальну кількість помилкових спрацьовувань, повне охоплення змін та низький час обробки сценаріїв (1–1,5 с). Це свідчить про адекватність та практичну придатність методу для підтримки управління кібербезпекою комп'ютерних мереж, а також про оптимальність його реалізації для реальних динамічних середовищ.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки для автоматизованого аналізу впливу змін у комп'ютерних мережах на аргументи кібербезпеки та отримано такі результати:

1. Проаналізовано сучасні підходи до забезпечення безпеки в комп'ютерних мережах та існуючі методи підтримки аргументів безпеки, включно з перевіркою узгодженості та формалізацією доказів.

2. Визначено сукупність типових шаблонів аргументів безпеки (контроль доступу, захист периметра, виявлення вторгнень, цілісність конфігурації) та сформульовано критерії, релевантні для оцінки їх узгодженості та коректності.

3. Розроблено формальну метамодель представлення аргументів безпеки, включно з механізмом семантичної простежуваності між елементами аргументів та артефактами системної інженерії, каталогом правил узгодженості та моделлю взаємозв'язків результатів перевірки та валідації безпеки.

4. Розроблено метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки для автоматизованого аналізу впливу змін на аргументи кібербезпеки, який забезпечує точне визначення узгодженості аргументів при зміні артефактів мережевої системи.

5. Реалізовано програмне забезпечення з підтримкою всіх чотирьох типових шаблонів аргументів безпеки, включно з GUI та базою даних для відстеження змін, проведено експериментальну перевірку методу та оцінено ефективність програмного забезпечення за метриками точності, часу аналізу, кількості помилкових спрацьовувань та охоплення змін.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Kaur J., Ramkumar K. The recent trends in cyber security: A review. *J. King Saud Univ.-Comput. Inf. Sci*. 2022. № 34. Pp. 5766–5781.

2. Shajan A., Rangaswamy S. Survey of security threats and countermeasures in cloud computing. *United Int. J. Res. Technol*. 2021, № 2. Pp. 201–207.

3. Salahdine F., Kaabouch N. Social Engineering Attacks: A Survey. *Future Internet.* 2019. № 11. P. 89.

4. Lu Y., Xu L.D. Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics. *IEEE Internet Things J.* 2019. № 6. Pp. 2103–2115.

5. Xiong W., Legrand E., Åberg O., Lagerström R. Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix. *Softw. Syst. Model*. 2021. № 21. Pp. 157–177.

7. Corallo A., Lazoi M., Lezzi M. Cybersecurity in the context of industry 4.0: A structured classification of critical assets and business impacts. *Comput. Ind*. 2020. № 114, 103165.

8. Bedratyuk L. and Savenko O., The star sequence and the general first Zagreb index, MATCH Communications in Mathematical and in Computer Chemistry. 2018. №-79. С. 407–414. https://doi.org/10.48550/arXiv.1706.00829

9. Khraisat A., Gondal I., Vamplew P., Kamruzzaman J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity*. 2019, 2, 20.

10. Tounsi W., Rais H. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Comput. Secur*. 2018, 72, P. 212–233.

11. Narang S. The reality of zero-day vulnerabilities. *Comput. Fraud Secur*. 2021, 20 p.

12. Dede G., Petsa A., Kavalaris S., Serrelis E., Evangelatos S., Oikonomidis I., and Kamalakis T. Cybersecurity as a contributor toward resilient Internet of Things (IoT) infrastructure and sustainable economic growth. *Information.* 2024. № 15: P. 798.

13. Rajasekar V. Premalatha J. Dhanaraj R.K. Security analytics. *In System Assurances; Elsevier: Amsterdam*, The Netherlands, 2022. Pp. 333–354.

14. Nallaperumal K. CyberSecurity Analytics to Combat Cyber Crimes. *In Proceedings of the 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Madurai, India, 13–15 December 2018. Pp. 1–4.

15. Khan S., Olivia T.S.L., Khan N., Why N.K., Wei T.S. Data Analytic for Cyber Security: A Review of Current Framework Solutions, Challenges and Trends. *Eurasia Proc. Sci. Technol. Eng. Math*. 2022. № 18, Pp. 1–6.

16. Verma R. Security Analytics: Adapting Data Science for Security Challenges. *In Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics, CODASPY '18, Tempe, AZ, USA, 19–21 March 2018*. Pp. 40–41.

17. Sharma G., Tyagi B. Security Analytics: Challenges and Future Directions. *IITM J. Manag. IT.* 2017. № 8. Pp. 37–41.

18. Cañizares J., Copeland S., Doorn N. Making Sense of Resilience. *Sustainability.* 2021. № 13, 8538.

19. Seth C,, Coravos A., Fahs G., Hatch A., Medina J., Woods B., Corman J. Building resilient medical technology supply chains with a software bill of materials. *Npj Digital Medicine 4*: 34. 2021.

20. Dunn C., Eriksen C., Scharte B. Making cyber security more resilient: Adding social considerations to technological fixes. *Journal of Risk Research 26*: 513–27. 2023.

21. Wesley C. 2022. Confidentiality, Integrity and Availability (CIA Triad). *Available online: https://www.techtarget.com/whatis/definition/Confidentiality-integrity-and-availability-CIA*.

22. Giorgio C. P. Understanding the Regulatory Approach of the Cyber Resilience Act: Protection of Fundamental Rights in Disguise? *European Journal of Risk Regulation 16*: 469–84. 2025.

23. Colabianchi S., Costantino F., Di Gravio G., Nonino F., Patriarca R. 2021. Discussing resilience in the context of cyber physical systems. *Computers & Industrial Engineering 160*: 107534.

24. Jing, X., Yan Z., Pedrycz W. Security Data Collection and Data Analytics in the Internet: A Survey. *IEEE Commun. Surv. Tutor*. 2019. № 21. Pp. 586–618.

25. Rassam M.A., Maarof M., Zainal A. Big Data Analytics Adoption for

Cybersecurity: A Review of Current Solutions, Requirements, Challenges and Trends. *J. Inf. Assur. Secur*. 2017. № 11. Pp. 124–145.

26. Birzniece I. Security Analytics: Dispelling the Fog. *In Proceedings of the BIR 2018 Short Papers, Workshops and Doctoral Consortium Co-Located with 17th International Conference Perspectives in Business Informatics Research (BIR 2018), Stockholm, Sweden, 24–26 September 2018.* Vol. 2218. P. 160–169.

27. Grahn K., Westerlund M., Pulkkis G. Analytics for Network Security: A Survey and Taxonomy. *In Information Fusion for Cyber-Security Analytics; Springer International Publishing: Cham, Switzerland*. 2016. P. 175–193.

28. Cremer F., Sheehan B., Fortmann M., Kia A.N., Mullins M., Murphy F. Materne S. Cyber risk and cybersecurity: A systematic review of data availability. Geneva Pap. *Risk Insur.-Issues Pract.* 2022. № 47. Pp. 698–736.

29. Marican M.N.Y., Razak S.A., Selamat A., Othman S.H. Cyber Security Maturity Assessment Framework for Technology Startups: A Systematic Literature Review. *IEEE Access.* 2023. № 11. Pp. 5442–5452.

30. Ratchford M., El-Gayar O., Noteboom C., Wang Y. BYOD security issues: A systematic literature review. *Inf. Secur. J. Glob. Perspect*. 2021. № 31. Pp. 253–273.

31. Garg M., Goel A. A systematic literature review on online assessment security: Current challenges and integrity strategies. *Comput. Secur*. 2022. № 113, 102544.

32. Abraham S., Nair S. Cyber Security Analytics: A Stochastic Model for Security Quantification Using Absorbing Markov Chains. *J. Commun*. 2014. № 9. Pp. 899–907.

33. Li X., Li J., Qu Y., He D. Semi-supervised gear fault diagnosis using raw vibration signal based on deep learning. *Chinese Journal of Aeronautics*. 2020. Vol. 33. P. 418–426.

34. Surindra M. D. et al. Use of machine learning models in condition monitoring of abrasive belt in robotic arm grinding process. *Journal of Intelligent Manufacturing*. 2025. Vol. 36. P. 3345–3358.

35. Li X. et al. Deep dynamic high-order graph convolutional network for wear fault diagnosis of hydrodynamic mechanical seal. *Reliability Engineering & System Safety*. 2024. Vol. 247. Art. 110117.

36. Dai J., Tian L., Han T., Chang H. Digital Twin for wear degradation of sliding bearing based on PFENN. *Advanced Engineering Informatics*. 2024. Vol. 61. Art. 102512.

37. Guo Y. et al. Research on Radial Rotor Plunger Wear Fault Monitoring Method by Fused Sound Vibration Signal Features. *IEEE Sensors Journal*. 2024. Vol. 24. P. 20896–20907.

38. Zhang M., Liu D., Liu Y. Recent progress in precision measurement and assembly optimization methods of the aero-engine multistage rotor: A comprehensive review. *Measurement*. 2024. Vol. 235. Art. 114990.

39. González-Granadillo G., González-Zarzosa S., Diaz R. Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures. *Sensors*. 2021. № 21. 4759.

40. Bahrami P.N., Dehghantanha A., Dargahi T., Parizi R.M., Choo K.K.R., Javadi H.H. Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tactics, techniques, and procedures. *J. Inf. Process. Syst*. 2019. № 15. Pp. 865–889.

41. Wen J. et al. Residual-based adversarial feature decoupling for remaining useful life prediction of aero-engines under variable operating conditions. *Expert Systems with Applications*. 2024. Vol. 255. Art. 124538.

42. Maurya M., Panigrahi I., Dash D., Malla C. Intelligent fault diagnostic system for rotating machinery based on IoT with cloud computing and artificial intelligence techniques: A review. *Soft Computing*. 2023. Vol. 28. P. 477–494.

43. Thamba N. B. et al. Comparison of ML Algorithms and Neural Networks on Fault Diagnosis of a Worm Gear. *Journal of Vibration Engineering & Technologies*. 2024. Vol. 12. P. 6355–6370.

44. Pulyala R. S., Gupta D. A., Dutt J. V. The Impact of Security Orchestration, Automation, and Response (SOAR) on Security Operations Center (SOC) Efficiency: A Comprehensive Analysis. *Turk. J. Comput. Math. Educ*. (TURCOMAT) 2019. № 10. Pp. 1545–1549.

45. Cinque M., Cotroneo D., Pecchia A. Challenges and Directions in Security Information and Event Management (SIEM). *In Proceedings of the 2018 IEEE*

International Symposium on Software Reliability Engineering Workshops (ISSREW), Memphis, TN, USA, 15–18 October 2018. Pp. 95–99.

46. Rosado D.G., Moreno J., Sánchez L.E., Santos-Olmo A., Serrano M.A., Fernández-Medina E. MARISMA-BiDa pattern: Integrated risk analysis for big data. *Comput. Secur.* 2021. № 102, 102155.

47. Jaeger D., Cheng F., Meinel C. Towards a system for complex analysis of security events in large-scale networks. *Comput. Secur*. 2017. № 67. Pp. 16–34.

48. Zou Q., Zhang L., Singhal A., Sun X., Liu P. Attacks on ML Systems: From Security Analysis to Attack Mitigation. *In Information Systems Security; Springer Nature: Cham, Switzerland,* 2022. Pp. 119–138.

49. Ulmer A., Schufrin M., Lücke-Tieke H., Kannanayikkal C.D., Kohlhammer J. Towards Visual Cyber Security Analytics for the Masses. *In Proceedings of the EuroVis Workshop on Visual Analytics 2018, Brno, Czech Republic, 4 June 2018*.

50. Geluvaraj B., Satwik P.M., Kumar, T.A. The Future of Cybersecurity: Major Role of Artificial Intelligence, Machine Learning, and Deep Learning in Cyberspace. *Electronics.* 2025. № 14. 2252.

51. Alani M.M. Big data in cybersecurity: A survey of applications and future trends. *J. Reliab. Intell. Environ*. 2021. № 7. Pp. 85–114.

52. Alavizadeh H., Alavizadeh H., Jang-Jaccard J. Cyber Situation Awareness Monitoring and Proactive Response for Enterprises on the Cloud. *In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020–1 January 2021*. Pp. 1276–1284.

53. Hussein M.K., Zainal B. N., Jaber A.N. Data security analysis for DDoS defense of cloud-based networks. *In Proceedings of the 2015 IEEE Student Conference on Research and Development (SCOReD), Kuala Lumpur, Malaysia, 13–14 December 2015*. Pp. 305–310.

54. Niu D.D., Liu L., Zhang X., Lü S., Li Z. Security analysis model, system architecture and relational model of enterprise cloud services. *Int. J. Autom. Comput*. 2016. № 13. Pp. 574–584.

55. Zhu G., Zeng Y., Guo M. A Security Analysis Method for Supercomputing Users' Behavior. *In Proceedings of the 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), New York, NY, USA, 26–28 June 2017*. Pp. 287–293.

56. Win T.Y., Tianfield H., Mair Q. Big Data Based Security Analytics for Protecting Virtualized Infrastructures in Cloud Computing. *IEEE Trans. Big Data.* 2018. № 4. Pp. 11–25.

57. Elsayed M.A., Zulkernine M. PredictDeep: Security Analytics as a Service for Anomaly Detection and Prediction. *IEEE Access*. 2020. № 8. Pp. 45184–45197.

58. Empl P., Pernul G. A Flexible Security Analytics Service for the Industrial IoT. *In Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems, CODASPY '21, Virtual, 28 April 2021*. Pp. 23–32.

59. Taylor T., Araujo F., Shu X. Towards an Open Format for Scalable System Telemetry. *In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December.* 2020. Pp. 1031–1040.

60. Sharma S., Sharma A., Saini H. Advanced Network Security Analysis (ANSA) in Big Data Technology. Int. J. Innov. *Technol. Explor. Eng*. 2019. № 8. Pp. 2634–2639.

61. Mystkowski A. et al. Measurement and diagnostic system for detecting and classifying faults in the rotary hay tedder using multilayer perceptron neural networks. *Engineering Applications of Artificial Intelligence*. 2024. Vol. 133. Art. 108513.

62. Jenab K., Khoury S., LaFevor K. Flow-Graph and Markovian Methods for Cyber Security Analysis. *Int. J. Enterp. Inf. Syst*. 2016. № 12. Pp. 59–84.

63. Valja M., Lagerstrom R,, Korman M., Franke U. Bridging the gap between business and technology in strategic decision-making for cyber security management. *In Proceedings of the 2016 Portland International Conference on Management of Engineering and Technology (PICMET), Honolulu, HI, USA, 4–8 September 2016*. Pp. 32–42.

64. Naik N., Jenkins P., Savage N., Katos V. Big data security analysis approach using Computational Intelligence techniques in R for desktop users. *In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece,*

6–9 December 2016. Pp. 1–8.

65. Wu L., Deng T. Computer Network Security Analysis Modeling Based on Spatio-temporal Characteristics and Deep Learning Algorithm. *J. Phys. Conf. Ser*. 2020, 1648, 042111.

66. An empirical study of intelligent security analysis methods utilizing big data. *J. Logist. Inform. Serv. Sci.* 2022. № 9. Pp. 26–35.

67. Gruber H., Fuchs A., Bader M. Evaluation of a Condition Monitoring Algorithm for Early Bearing Fault Detection. *Sensors.* 2024. Vol. 24. Art. 2138.

68. Kumar R., Singh S., Kela R. A Quantitative Security Risk Analysis Framework for Modelling and Analyzing Advanced Persistent Threats. *In Foundations and Practice of Security; Springer International Publishing: Cham, Switzerland*. 2021. Pp. 29–46.

69. Moshika A., Thirumaran M., Natarajan B., Andal K., Sambasivam G., Manoharan R. Vulnerability Assessment in Heterogeneous Web Environment Using Probabilistic Arithmetic Automata. *IEEE Access 2021*. № 9. Pp. 74659–74673.

70. Lagerstrom R., Johnson P., Ekstedt M. Automatic Design of Secure Enterprise Architecture: Work in Progress Paper. *In Proceedings of the 2017 IEEE 21st International Enterprise Distributed Object Computing Workshop (EDOCW), Quebec, QC, Canada, 10–13 October 2017*. Pp. 65–70.

71. Yang F., Tian X., Ma L., Shi X. An optimized variational mode decomposition and symmetrized dot pattern image characteristic information fusion-Based enhanced CNN ball screw vibration intelligent fault diagnosis approach. *Measurement*. 2024. Vol. 229. Art. 114382.

72. Ahmed A., Hameed S., Rafi M., Mirza Q.K.A. An Intelligent and Time-Efficient DDoS Identification Framework for Real-Time Enterprise Networks: SAD-F: Spark Based Anomaly Detection Framework. *IEEE Access 2020*. № 8. Pp. 219483–219502.

73. Padmanaban R., Thirumaran M., Sanjana V., Moshika A. Security Analytics for Heterogeneous Web. *In Proceedings of the 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 29–30 March 2019*. Pp. 1–6.

74. Ndichu S., Ban T., Takahashi T., Inoue D. Critical-Threat-Alert Detection using Online Machine Learning. In Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2022; pp. 3007–3014.

75. Efiong J.E., Akinyemi B.O., Olajubu E.A., Aderounmu G.A., Degila J. CyberSCADA Network Security Analysis Model for Intrusion Detection Systems in the Smart Grid. *In Advances in Intelligent Systems, Computer Science and Digital Economics IV; Springer Nature: Cham, Switzerland*. 2023. Pp. 481–499.

76. Chowdhary A., Huang D., Mahendran J.S., Romo D., Deng, Y., Sabur A. Autonomous Security Analysis and Penetration Testing. *In Proceedings of the 2020 16th International Conference on Mobility, Sensing and Networking (MSN), Tokyo, Japan, 17–19 December 2020*. Pp. 508–515.

77. Sundararaj A., Knittl S., Grossklags J. Challenges in IT Security Processes and Solution Approaches with Process Mining. *In Security and Trust Management; Springer International Publishing: Cham, Switzerland*. 2020. Pp. 123–138.

78. Aquino M.F.M., Noroña M.I. Enhancing cyber security in the Philippine academe: A risk-based it project assessment approach. *In Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management, Singapore, 7–11 March 2021*. Pp. 5166–5179.

79. Chen G., Mazin T. Computer Network Security Analysis Based on Deep Learning Algorithm. *In Application of Intelligent Systems in Multi-Modal Information Analytics; Springer International Publishing: Cham, Switzerland*. 2022. Pp. 993–998.

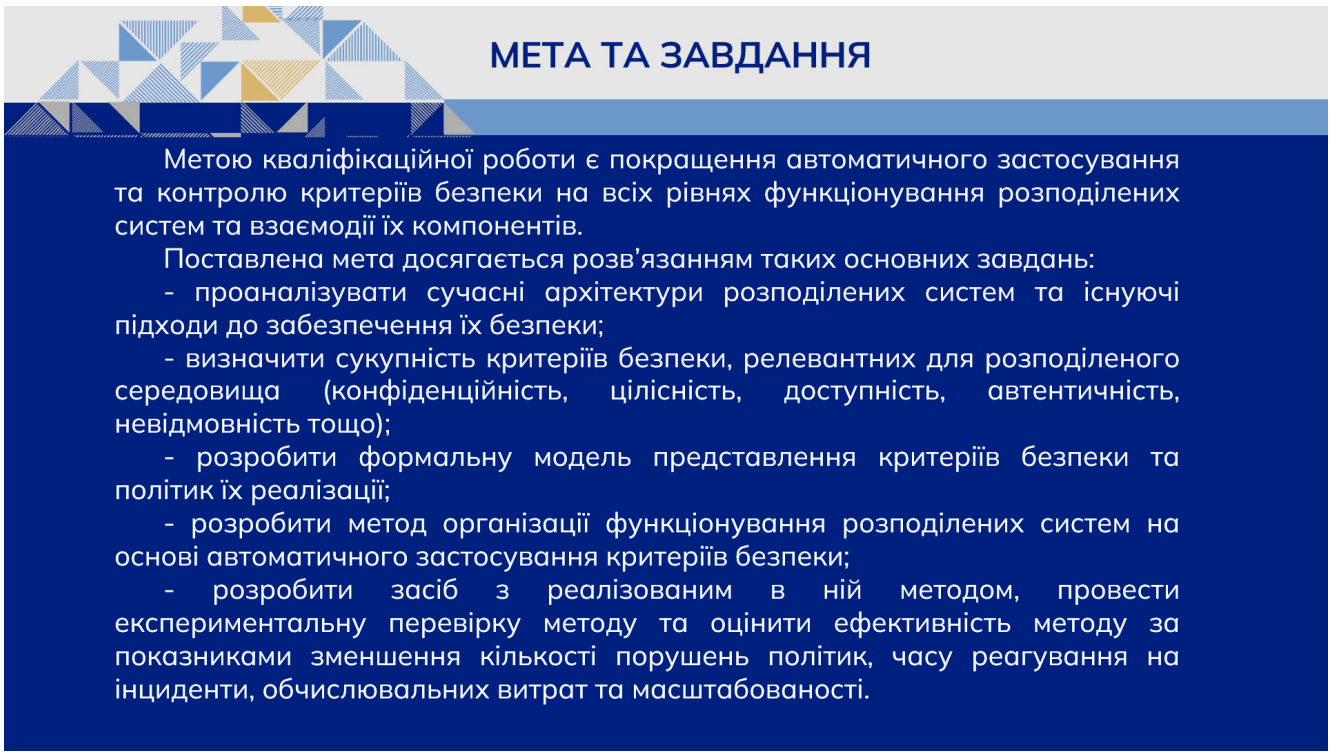
80. Melnychenko O., Savenko O., Radiuk P. Apple Detection with Occlusions Using Modified YOLOv5-v1. *IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems:Technology and Applications (IDAACS)*, Dortmund, Germany, 2023, pp. 107-112, doi: 10.1109/IDAACS58523.2023.10348779

81. Кушнір Д., Регіда П., Клейн О., Віжевський П. Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки. *Вимірювальна та обчислювальна техніка в технологічних процесах.* 2026. №2.

Додаток А

(обов'язковий)

Презентація до роботи



ОБ'ЄКТ, ПРЕДМЕТ

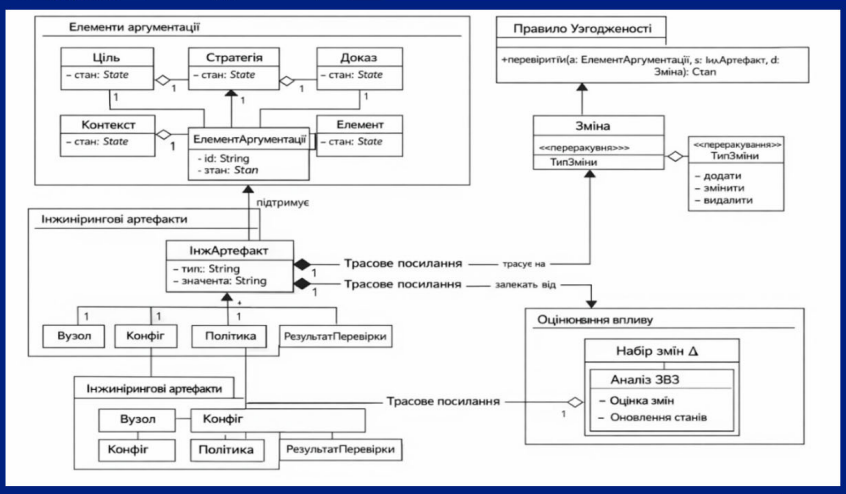
Об'єктом дослідження є процеси функціонування та автоматизації управління безпекою у розподілених системах.

Предметом дослідження є методи, моделі та засоби автоматизованого застосування формалізованих критеріїв безпеки під час функціонування розподілених Наукова новизна отриманих результатів:

систем.

- новий метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки, який, на відміну від відомих підходів, що базуються на статичному налаштуванні політик та ручному адмініструванні засобів захисту, забезпечує інтегроване та формалізоване управління безпекою на всіх рівнях взаємодії компонентів системи, надає системний механізм керованості, передбачуваності та стійкості розподіленої інфраструктури за рахунок вбудованої, автоматизованої та формалізованої моделі забезпечення безпеки, що функціонує в реальному часі та масштабується разом із системою.

МЕТАМОДЕЛЬ АРГУМЕНТІВ БЕЗПЕКИ



$$M = (A, R, T, C)$$

КЛАСИ ТИПОВИХ ШАБЛОНІВ АРГУМЕНТІВ БЕЗПЕКИ

Приклади типових шаблонів аргументів безпеки для комп'ютерних мереж.

1. Шаблон аргументу безпеки для контролю доступу використовується для обґрунтування того, що доступ до ресурсів комп'ютерної мережі здійснюється лише авторизованими користувачами відповідно до встановленої політики безпеки.

2. Шаблон аргументу безпеки для захисту мережного периметра застосовується для підтвердження того, що мережа захищена від несанкціонованого доступу із зовнішніх мереж за допомогою міжмережних екранів та механізмів фільтрації трафіку.

3. Шаблон аргументу безпеки для виявлення вторгнень використовується для доведення здатності системи виявляти та реєструвати спроби несанкціонованого доступу або мережні атаки.

4. Шаблон аргументу безпеки для забезпечення цілісності конфігурації призначений для підтвердження того, що конфігурації мережних пристроїв та програмних компонентів не змінюються без контролю та відповідної авторизації.

КРОКИ МЕТОДУ

1.Ініціалізація моделей та завантаження вихідних даних.

2.Виявлення та формалізація змін у мережі.

3.Застосування механізму семантичної простежуваності.

4.Визначення області впливу змін.

5.Перевірка узгодженості аргументів безпеки.

6.Аналіз взаємозв'язків результатів перевірки та валідації.

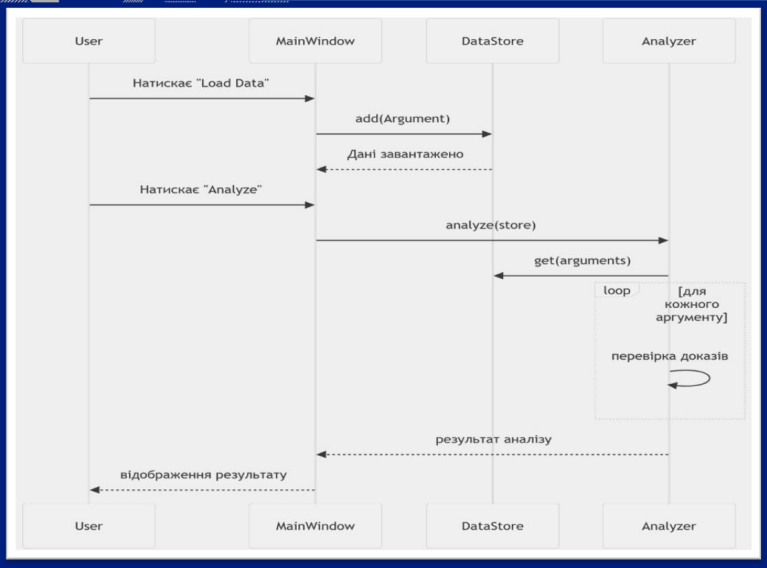
7.Оцінювання повноти та достовірності аргументації.

8.Класифікація впливу змін.

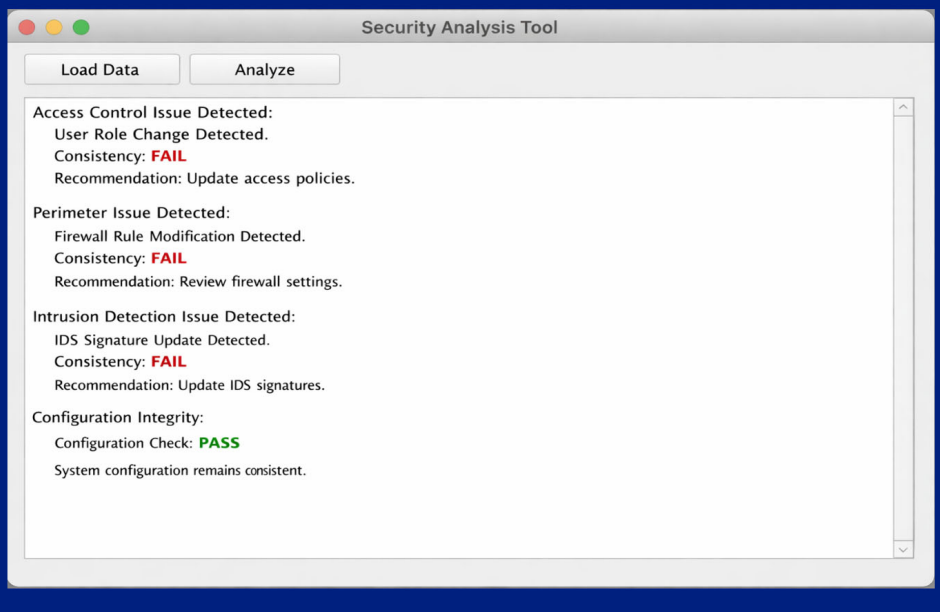
9.Формування рекомендацій щодо оновлення аргументів безпеки.

10.Оновлення аргументів безпеки та повторна перевірка.

UML-ДІАГРАМА ПОСЛІДОВНОСТЕЙ



ВІКОННА ФОРМА ІНТЕРФЕЙСУ КОРИСТУВАЧА



РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТУ

Таблиця 1

Метрики точності та узгодженості аргументів безпеки

Тип шаблону аргументу	Кількість сценаріїв	Узгоджені аргументи (виявлені)	Неузгоджені аргументи (виявлені)	Accuracy (%)	False Positives	False Negatives
Контроль доступу	5	12	8	96	1	0
Захист периметра	5	10	10	94	1	1
Виявлення вторгнень	5	11	9	95	0	1
Цілісність конфігурації	5	14	6	97	0	0

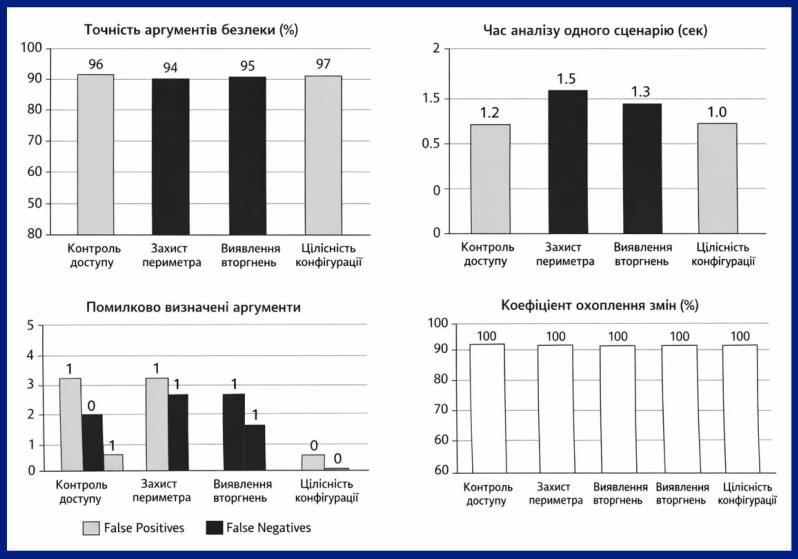
РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТУ

Таблиця 2

Метрики часу аналізу

Тип шаблону аргументу	Середній час аналізу одного сценарію (сек)	Коефіцієнт охоплення змін (%)
Контроль доступу	1.2	100
Захист периметра	1.5	100
Виявлення вторгнень	1.3	100
Цілісність конфігурації	1.0	100

РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТУ



ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки для автоматизованого аналізу впливу змін у комп'ютерних мережах на аргументи кібербезпеки та отримано такі результати:

- проаналізовано сучасні підходи до забезпечення безпеки в комп'ютерних мережах та існуючі методи підтримки аргументів безпеки, включно з перевіркою узгодженості та формалізацією доказів;
- визначено сукупність типових шаблонів аргументів безпеки (контроль доступу, захист периметра, виявлення вторгнень, цілісність конфігурації) та сформульовано критерії, релевантні для оцінки їх узгодженості та коректності;
- розроблено формальну метамодель представлення аргументів безпеки, включно з механізмом семантичної простежуваності між елементами аргументів та артефактами системної інженерії, каталогом правил узгодженості та моделлю взаємозв'язків результатів перевірки та валідації безпеки;
- розроблено метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки для автоматизованого аналізу впливу змін на аргументи кібербезпеки, який забезпечує точне визначення узгодженості аргументів при зміні артефактів мережевої системи;
- реалізовано програмне забезпечення з підтримкою всіх чотирьох типових шаблонів аргументів безпеки, включно з GUI та базою даних для відстеження змін, проведено експериментальну перевірку методу та оцінено ефективність програмного забезпечення за метриками точності, часу аналізу, кількості помилкових спрацьовувань та охоплення змін.

Додаток Б

(обов'язковий)

Наукова праця здобувача

УДК 004.7:004.3

КУШНІР Дмитро
Хмельницький національний університет
ORCID ID: 0009-0001-6581-6081
e-mail: dima99.kushnir@gmail.com
РЕГІДА Павло
Хмельницький національний університет
ORCID ID: 0000-0002-6591-7069
e-mail: regidap@khmnu.edu.ua
КЛЕЙН Олександр
Хмельницький національний університет
ORCID ID: 0000-0002-1896-943X
e-mail: olexandrkleyn@gmail.com
ВІЖЕВСЬКИЙ Петро
Хмельницький національний університет
ORCHID ID: 0009-0009-4851-0839
e-mail: vizhevskyipv@khmnu.edu.ua

МЕТОД ОРГАНІЗАЦІЇ ФУНКЦІОНУВАННЯ РОЗПОДІЛЕНИХ СИСТЕМ НА ОСНОВІ АВТОМАТИЧНОГО ЗАСТОСУВАННЯ КРИТЕРІЇВ БЕЗПЕКИ

Сучасний розвиток інформаційних технологій супроводжується активним переходом до розподілених обчислювальних архітектур, зокрема хмарних середовищ, мікросервісних систем, контейнеризованих інфраструктур та edge-обчислень. Такі підходи забезпечують високу масштабованість, гнучкість і відмовостійкість, проте суттєво ускладнюють забезпечення інформаційної безпеки через динамічність топології, значну кількість взаємодіючих компонентів і різнорівневу довіру між ними. У цих умовах традиційні методи, що базуються на статичних політиках і ручному адмініструванні, виявляються недостатньо ефективними, що зумовлює необхідність автоматизації процесів забезпечення безпеки.

У роботі запропоновано метод організації функціонування розподілених систем на основі автоматизованого застосування формалізованих критеріїв безпеки, який орієнтований на аналіз впливу змін у комп'ютерних мережах на аргументи кібербезпеки. Метод передбачає використання метамоделі аргументів безпеки, множини типових шаблонів, механізму семантичної простежуваності, формалізованого каталогу правил узгодженості та моделі взаємозв'язків результатів перевірки і валідації. Це дозволяє забезпечити інтеграцію вимог безпеки у структуру системи, автоматизувати контроль їх виконання та своєчасно виявляти невідповідності.

Розроблене програмне забезпечення реалізує запропонований метод і забезпечує аналіз чотирьох типових шаблонів аргументів безпеки: контроль доступу, захист мережевого периметра, виявлення вторгнень і цілісність конфігурації. Експериментальна перевірка показала високу точність визначення узгодженості аргументів (94–97%), мінімальну кількість помилкових спрацьовувань, повне охоплення змін та низький час обробки сценаріїв (1–1,5 с). Отримані результати підтверджують адекватність і практичну придатність методу для використання у динамічних розподілених середовищах.

Подальші дослідження доцільно спрямувати на розширення та деталізацію моделей для різних типів аргументів безпеки та підвищення рівня їх адаптивності.

Ключові слова: розподілена система, комп'ютерна система, комп'ютерна мережа, автоматизація, критерії безпеки, показники компроментації, кібербезпека, захист периметру, контроль доступу.

KUSHNIR Dmytro
REHIDA Pavlo
KLEIN Olexandr
VIZHEVSKYI Petro
Khmelnytskyi National University, Khmelnytskyi, Ukraine

METHOD OF ORGANIZING THE FUNCTIONING OF DISTRIBUTED SYSTEMS BASED ON THE AUTOMATIC APPLICATION OF SECURITY CRITERIA

The modern development of information technologies is accompanied by an active transition to distributed computing architectures, in particular cloud environments, microservice systems, containerized infrastructures and edge computing. Such approaches provide high scalability, flexibility and fault tolerance, but significantly complicate the provision of information security due to the dynamics of the topology, a significant number of interacting components and different levels of trust between them. In these conditions, traditional methods based on static policies and manual administration are not effective enough, which necessitates the automation of security processes.

The paper proposes a method of organizing the functioning of distributed systems based on the automated application of formalized security criteria, which is focused on analyzing the impact of changes in computer networks on cyber security arguments. The method involves the use of a metamodel of security arguments, a set of typical templates, a mechanism of semantic traceability, a formalized catalog of consistency rules, and a model of the relationships of verification and validation results. This makes it possible to ensure the integration of security requirements into the structure of the system, to automate the control of their implementation, and to detect inconsistencies in a timely manner.

The developed software implements the proposed method and provides analysis of four typical patterns of security arguments: access control, network perimeter protection, intrusion detection, and configuration integrity. Experimental verification showed high accuracy in determining the consistency of arguments (94–97%), minimal number of false positives, full coverage of changes and low processing time of scenarios (1–1.5 s). The obtained results confirm the adequacy and practical suitability of the method for use in dynamic distributed environments.

Further research should be focused on expanding and detailing models for different types of security arguments and increasing their adaptability.

Keywords: distributed system, computer system, computer network, automation, security criteria, indicators of compromise, cyber security, perimeter protection, access control.

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ

Сучасний етап розвитку інформаційних технологій характеризується стрімким переходом до розподілених обчислювальних архітектур, зокрема хмарних платформ, мікросервісних середовищ, контейнеризованих інфраструктур та edge-обчислень. Такі системи забезпечують масштабованість, відмовостійкість та гнучкість, однак водночас істотно ускладнюють забезпечення інформаційної безпеки.

На відміну від централізованих систем, розподілені середовища мають динамічну топологію, велику кількість взаємодіючих вузлів, асинхронні комунікації та різнорівневу довіру між компонентами. Це призводить до зростання кількості потенційних вразливостей, ускладнення контролю доступу, моніторингу подій безпеки та своєчасного реагування на інциденти.

Традиційні підходи до забезпечення безпеки, що базуються на ручному адмініструванні політик і статичних правилах контролю доступу, виявляються недостатньо ефективними в умовах динамічних розподілених систем. Людський фактор, складність конфігурацій та масштаб інфраструктури підвищують ризик помилок і невідповідностей політик безпеки.

Актуальним напрямом досліджень є розроблення методів організації функціонування розподілених систем, що передбачають автоматизоване застосування формалізованих критеріїв безпеки. Такий підхід дозволяє інтегрувати вимоги безпеки безпосередньо в архітектуру системи, забезпечити їх постійний контроль, автоматичну верифікацію та адаптивне коригування.

Автоматизація застосування критеріїв безпеки передбачає формалізацію політик доступу, використання моделей довіри, механізмів контролю взаємодій між компонентами, а також впровадження інструментів моніторингу та оркестрації безпекових процесів.

Таким чином, розроблення методу організації функціонування розподілених систем на основі автоматизованого застосування критеріїв безпеки є актуальною науково-практичною задачею, спрямованою на підвищення стійкості, надійності та керованості сучасних інформаційних інфраструктур.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Стійкість розподілених систем у сучасній парадигмі трактується як багатовимірна інтегральна характеристика, що поєднує інформаційну безпеку, функційну надійність, відмовостійкість, адаптивність та керованість у динамічному середовищі. Вона визначає здатність системи зберігати коректність стану, неперервність виконання логічних операцій і допустимий рівень сервісу за умов дії внутрішніх збоїв, мережних порушень, навмисних атак та невизначеності часових параметрів. Разом із тим, незважаючи на значний обсяг досліджень у галузі теорії розподілених обчислень, проблема комплексного забезпечення стійкості залишається остаточно не розв'язаною, що зумовлено фундаментальними алгоритмічними обмеженнями та зростаючою складністю сучасних інфраструктур [1, 2].

Відомі підходи до забезпечення стійкості зосереджені переважно на окремих аспектах, які включають реплікації даних, досягнення консенсусу, резервування ресурсів або впровадженні криптографічних механізмів захисту. Алгоритми консенсусу дозволяють забезпечити узгодженість стану в умовах відмов частини вузлів; проте вони супроводжуються істотними витратами обчислювальних і мережних ресурсів та не вирішують проблему динамічного конфлікту політик безпеки. Теорія відмовостійкості пропонує моделі активного та пасивного резервування, але не гарантує автоматичної узгодженості безпекових критеріїв у масштабі всієї системи. Криптографічні механізми забезпечують цілісність і автентичність повідомлень, однак не усувають ризиків логічної неконсистентності або накопичення прихованих конфігураційних помилок [3, 4].

Фундаментальним обмеженням функційної стійкості залишається компроміс, окреслений теоремою CAP [5, 6], відповідно до якої в умовах мережного поділу неможливо одночасно гарантувати строгі властивості узгодженості, доступності та толерантності до розділення мережі. Існуючі архітектурні рішення змушені обирати між послабленими моделями узгодженості або зниженням доступності сервісу, що створює простір для потенційних зловживань і порушень цілісності. Невирішеною залишається проблема [7, 8] формалізованого автоматичного вибору оптимального режиму функціонування залежно від поточного стану мережі та рівня загроз.

Окремої уваги потребує проблема самостабілізації розподілених систем. Хоча теоретичні моделі самостабілізуючих алгоритмів доводять можливість повернення системи до легітимного стану з довільної конфігурації, практична інтеграція таких механізмів у великомасштабні інфраструктури ускладнена через відсутність формалізованих критеріїв легітимності та складність глобальної верифікації стану. Залишається відкритим питання побудови універсального механізму, здатного автоматично ідентифікувати деградаційні процеси та ініціювати процедури реконфігурації без централізованого контролю [9, 10].

Невирішеною також є проблема латентних прихованих відмов, які не призводять до повної зупинки вузла, але поступово спотворюють результати обчислень або накопичують помилкові стани. Традиційні механізми моніторингу орієнтовані на явні відмови та перевищення порогових значень, тоді як складні багатовекторні атаки або логічні конфлікти політик можуть залишатися непоміченими тривалий час. Відсутність формальної моделі узгодженості безпекових критеріїв у масштабі всієї системи унеможливлює гарантоване виявлення таких аномалій [11, 12].

Складність забезпечення стійкості зростає в асинхронних мережах, де неможливо достовірно відрізнити відмову вузла від його тимчасової затримки. Це породжує ризики розгалуження стану, появи суперечливих транзакцій та втрати глобальної консистентності. Існуючі протоколи частково вирішують цю проблему шляхом введення тайм-аутів або обрання лідера, однак такі підходи залишаються вразливими до навмисних затримок та атак типу «відмова в обслуговуванні» [13, 14].

В роботах [15, 16] запропоновано підходи до забезпечення стійкості розподіленої інфраструктури в умовах впливів комп'ютерних атак.

Таким чином, можна констатувати, що в сучасних дослідженнях вирішено низку часткових задач таких, як забезпечення реплікації стану, формалізацію алгоритмів консенсусу, побудову моделей відмовостійкості, застосування криптографічних засобів підтвердження цілісності та розроблення механізмів резервування. Водночас комплексна проблема [17, 18] інтеграції функційної стійкості та інформаційної безпеки в єдину формалізовану систему автоматичного управління залишається відкритою.

Перспективними дослідженнями [19, 20, 21] є такі, що полягають в обґрунтуванні підходу, за якого стійкість розглядається як динамічний процес розподіленого середовища, який підтримується автоматичним застосуванням формалізованих критеріїв безпеки до кожної взаємодії між компонентами системи. Запропонована концепція передбачає інтеграцію алгоритмічної надійності, політик доступу, механізмів виявлення конфліктів і процедур реконфігурації в єдину модель управління, що функціонує в реальному часі. У межах такого підходу потенційно можуть бути вирішені проблеми автоматичного узгодження політик, виявлення їх конфліктності, мінімізації ризику логічної неконсистентності, а також адаптивного вибору режиму функціонування системи залежно від рівня загроз та стану мережної інфраструктури. Це дозволяє перейти від реактивної моделі захисту до проактивної адаптивної архітектури, у якій стійкість забезпечується не лише надмірністю ресурсів, а й інтелектуальним керуванням поведінкою компонентів.

Сучасний стан досліджень характеризується наявністю ефективних локальних рішень, але відсутністю універсальної методології, що поєднує алгоритмічну узгодженість, безпекові критерії та механізми самовідновлення в єдину формалізовану систему. Розроблення такого підходу становить актуальну наукову

проблему та визначає напрям подальших досліджень у галузі організації функціонування стійких розподілених систем.

Метою роботи є покращення автоматичного застосування та контролю критеріїв безпеки на всіх рівнях функціонування розподілених систем та взаємодії їх компонентів.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Метамодель аргументів безпеки

Для автоматизованого аналізу впливу змін у комп'ютерних мережах на аргументи кібербезпеки потрібно розробити метод, який забезпечуватиме підтримку актуальності та узгодженості доказів безпечності системи при зміні її конфігурації або структури. Для реалізації такого методу необхідно сформувати формалізовану модель представлення аргументів безпеки, що дозволить описувати взаємозв'язки між елементами аргументації, вимогами безпеки, доказами перевірки та компонентами мережної інфраструктури. У процесі дослідження необхідно систематизувати та сформувати каталог правил узгодженості між елементами аргументів безпеки та артефактами комп'ютерної мережі, такими як конфігурації вузлів, мережні служби, політики доступу, програмні модулі та результати перевірок безпеки. Ці правила повинні визначати, яким чином зміни у структурі або параметрах мережної системи впливають на відповідні елементи аргументації безпеки, а також дозволяти автоматично визначати елементи, які можуть втратити актуальність або потребують повторної перевірки. Крім того, необхідно визначити та реалізувати механізм семантичної простежуваності між моделями системної інженерії комп'ютерної мережі та моделями аргументів безпеки. Такий механізм повинен забезпечувати встановлення формальних зв'язків між елементами мережної інфраструктури та твердженнями аргументів безпеки, що дозволить автоматично відстежувати зміни у системі та визначати їх вплив на відповідні елементи аргументації. Це, у свою чергу, забезпечить можливість автоматизованого оновлення статусу аргументів безпеки при модифікації мережних компонентів або їх конфігурацій.

Важливим завданням дослідження є також формалізація взаємозв'язків між різними результатами перевірки та валідації безпеки, які використовуються як докази під час формування аргументів безпеки. Необхідно визначити правила взаємозалежності між різними типами доказів, зокрема результатами статичного аналізу, тестування безпеки, перевірок конфігурацій та інших методів оцінювання захищеності. Це дозволить виконувати автоматизований аналіз повноти та достатності доказів безпечності системи, а також визначати вплив додавання, модифікації або видалення таких доказів на загальну структуру аргументації безпеки.

Для забезпечення практичної реалізації запропонованого підходу необхідно розробити структуру повторно використовуваних елементів аргументації безпеки, які можуть бути застосовані як типові шаблони або будівельні блоки під час формування аргументів безпеки комп'ютерних мереж. Для кожного такого елемента повинні бути визначені відповідні правила узгодженості та умови їх перевірки у разі зміни стану мережної системи.

Результатом виконання зазначених завдань має стати формування узагальненого методу організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки для автоматизованого аналізу впливу змін у комп'ютерних мережах на аргументи кібербезпеки, який забезпечуватиме можливість точного визначення порушень узгодженості між елементами аргументації безпеки, мережними артефактами та доказами перевірки безпеки, а також сприятиме підвищенню ефективності процесів управління змінами та підтримки актуальності аргументів безпеки у складних мережних системах.

Таким чином, потрібно здійснити розроблення метамоделі аргументів безпеки, множини типових шаблонів аргументів безпеки, механізм семантичної простежуваності кореляції між змінами, формалізований каталог правил узгодженості, формальна модель взаємозв'язків між результатами перевірки та валідації безпеки і метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки.

Спочатку розробимо метамодель аргументів безпеки сумісну зі стандартом GSN (Goal Structuring Notation), яка підтримує автоматизовану перевірку узгодженості та аналіз змін. Необхідно розробити метамодель аргументів безпеки, сумісну зі стандартом GSN, яка забезпечуватиме формалізоване представлення структури аргументів кібербезпеки комп'ютерних мереж та підтримуватиме автоматизовану перевірку їх узгодженості. Запропонована метамодель повинна визначати основні елементи аргументації безпеки, зокрема цілі безпеки, стратегії доведення, твердження, контексти та докази, а також описувати формальні зв'язки між ними.

У межах розроблення метамоделі необхідно передбачити механізми встановлення прямих зв'язків між елементами аргументів безпеки та артефактами комп'ютерної мережі, такими як компоненти мережної інфраструктури, конфігураційні параметри, політики безпеки, результати перевірок і тестування. Це дозволить забезпечити семантичну простежуваність між моделями системної інженерії мережі та структурою аргументів безпеки.

Крім того, метамодель повинна підтримувати можливість анотування зв'язків між елементами аргументації та системними артефактами спеціалізованими правилами узгодженості, що визначають умови коректності аргументів безпеки у разі зміни параметрів або структури мережної системи. На основі таких правил має бути забезпечена автоматизована перевірка узгодженості аргументів безпеки та визначення їх актуальності після внесення змін до конфігурації комп'ютерної мережі.

Метамодель аргументів безпеки представимо у вигляді формальної структури так:

$$M = (A, R, T, C), \tag{1}$$

де A - множина елементів аргументації безпеки; R - множина відношень між елементами аргументації; T - множина трасових зв'язків між аргументами безпеки та артефактами системи; C - множина правил узгодженості.

Множину елементів аргументів безпеки визначимо так:

$$A = G \cup S \cup E \cup K, \tag{2}$$

де G - множина цілей безпеки (Goals); S - множина стратегій доведення (Strategies); E - множина доказів або результатів перевірок (Evidence); K - множина контекстних елементів (Context).

Таким чином, задамо множину елементів аргументів безпеки так:

$$A = \{a_1, a_2, \dots, a_{n_A}\},\ a_i = (a_{i,id}, a_{i,type}, a_{i,state}), \tag{3}$$

де a_i - i-тий елемент аргументації, що описується кортежем; $i = 1,2, \dots, n_A$; n_A – кількість елементів множини A; $a_{i,id}$ - ідентифікатор елемента; $a_{i,type} \in G \cup S \cup E \cup K$; $a_{i,state}$ - стан узгодженості елемента.

Множина станів $A_{state} = \{a_{1,state}, a_{2,state}, \dots, a_{n_A,state}\}$, де $a_{i,state} \in$ {дійсний, непослідовний, застарілий}.

Відношення аргументації визначимо множиною так:

$$B \subseteq A \times A, \tag{4}$$

де елемент $(a_i, a_j) \in B$ і означає, що він підтримує або деталізує елемент a_i.

Функцію підтримки для елементів з формули (2.4) визначимо так:

$$F_{support}: A \to 2^A,\ F_{support}(a_i) = \{a_j | (a_i, a_j) \in B\}, \tag{5}$$

де елемент $(a_i, a_j) \in B$ і означає, що він підтримує або деталізує елемент a_i; B - відношення аргументації; A - множина елементів аргументів безпеки; 2^A – множина підмножин множини A.

Модель системних артефактів комп'ютерної мережі задамо через артефакти системи множиною так:

$$S_{sys} = \{s_1, s_2, \dots, s_{n_{S_{sys}}}\}, \tag{6}$$

де $s_j = (s_{j,type}, s_{j,value})$; $n_{S_{sys}}$ – кількість елементів множини S_{sys}; $j = 1,2, \dots, n_{S_{sys}}$.

Типи артефактів можуть включати вузли мережі, конфігурації, політики безпеки, результати тестування, програмні компоненти.

Зв'язок між аргументами безпеки та артефактами системи визначимо так:

$$T \subseteq A \times S_{sys}, \tag{7}$$

де елемент $(s_i, s_j) \in T$ і означає, що аргумент безпеки a_i залежить від артефакта системи s_j.

Функцію залежності визначимо так:

$$F_{trace}(a_i) = \{a_j | (a_i, s_j) \in T\}, \tag{8}$$

де елемент $(a_i, s_j) \in T$.

Зміни системи визначимо множиною так:

$$D_\Delta = \{(\delta, s_j) |\ j = 1,2, \dots, n_{S_{sys}}\}, \tag{9}$$

де s_j – змінений артефакт; δ - тип зміни; $\delta \in$ {доповнено, видалено, модифіковано}.

Правило узгодженості визначимо як функцію так:

$$c_k: A \times S_{sys} \times D_\Delta \to M_{state}, \tag{10}$$

де M_{state} = {дійсний, непослідовний, застарілий}; M_{state} – множина станів; D_Δ - множина змін системи; A - множина елементів аргументів безпеки; S_{sys} – множина артефактів системи.

Функція c_k визначає новий стан аргументу безпеки після зміни системного артефакта.

Стан елемента аргументації визначатимемо функцією так:

$$F_{state} = \begin{cases} \text{дійсний, якщо } c_k(a_i, s_j, \delta) = \text{дійсний}; \\ \text{непослідовний, якщо } c_k(a_i, s_j, \delta) = \text{непослідовний}; \\ \text{застаріший, якщо } c_k(a_i, s_j, \delta) = \text{застарілий}, \end{cases} \tag{11}$$

де $s_j = (s_{j,type}, s_{j,value})$; $n_{S_{sys}}$ – кількість елементів множини S_{sys}; $j = 1,2, \dots, n_{S_{sys}}$; a_i - i-тий елемент аргументації, що описується кортежем; $i = 1,2, \dots, n_A$; n_A – кількість елементів множини A; δ - тип зміни; $\delta \in$ {доповнено, видалено, модифіковано}.

Аналіз впливу змін задамо функцією так:

$$F_v(M, D_\Delta) \to A', \tag{12}$$

де M - метамодель аргументів безпеки; D_Δ - множина змін; A' - оновлений стан аргументів.

Суть визначення функції F_v полягає у визначенні змінених артефактів s_j та знаходженні всіх аргументів, тобто формування множини так:

$$A_s = \{a_i | (a_i, s_j) \in T\}, \tag{13}$$

де елемент $(a_i, s_j) \in T$.

Формула (13) додатково доповнює формулу (8) і далі потрібно застосувати правила узгодженості з множини правил узгодженості C та оновити стан аргументів.

Таким чином, згідно введених понять та співвідношень (1)-(13) задамо концептуальну метамодель аргументів безпеки комп'ютерної мережі так:

$$M_k = (A, R, S_{sys}, T, C), \tag{14}$$

де A - множина елементів аргументації безпеки; R - множина відношень між елементами аргументації; S_{sys} - артефакти мережної системи; T - множина трасових зв'язків між аргументами безпеки та артефактами системи; C - множина правил узгодженості.

Запропонована метамодель аргументів безпеки, узгоджена з підходом **Goal Structuring Notation,** формує цілісну формалізовану основу для автоматизованої підтримки аргументів кібербезпеки у комп'ютерних мережах. Її ключова особливість полягає у поєднанні трьох раніше роз'єднаних компонентів: структури аргументації безпеки; моделей системних (мережних) артефактів; механізму аналізу впливу змін. На відміну від традиційного використання GSN як графічного засобу документування аргументів, запропонована метамодель розширює його до рівня формальної, машинозчитуваної структури з чітко визначеними трасовими зв'язками та правилами узгодженості.

Особливістю метамоделі є введення семантичної простежуваності між елементами аргументів безпеки та артефактами комп'ютерної мережі. Це дозволяє перейти від статичного опису аргументації до динамічної моделі, здатної реагувати на зміни конфігурації системи. У результаті аргументи безпеки перестають бути лише текстовими або графічними обґрунтуваннями і набувають властивостей формальної моделі, що підтримує автоматизовану перевірку узгодженості. Таким чином, метамодель забезпечує можливість мінімізувати ручний аналіз під час управління змінами, підвищити точність визначення впливу змін та зменшити ризик використання застарілих доказів безпечності.

Практична цінність метамоделі полягає в тому, що вона дозволяє автоматично визначати аргументи безпеки, на які вплинули зміни у мережній інфраструктурі, формалізувати правила узгодженості між вимогами безпеки, доказами та системними компонентами, оцінювати повноту доказової бази після модифікації конфігурації системи, забезпечувати контроль актуальності аргументів безпеки в процесі експлуатації мережі. Таким чином, метамодель створює основу для побудови інтелектуальних інструментів підтримки процесів забезпечення кібербезпеки, аудиту та сертифікації мережних систем.

Отже, запропонована метамодель є теоретично обґрунтованою основою для автоматизованого аналізу впливу змін на аргументи кібербезпеки комп'ютерних мереж. Вона забезпечує формалізацію структури аргументації, інтеграцію з моделями системної інженерії та підтримку автоматизованої перевірки узгодженості. Подальше розширення та інструментальна реалізація метамоделі дозволять створити повноцінний метод підтримки актуальності аргументів безпеки у динамічних мережних середовищах.

Механізм семантичної простежуваності кореляції між змінами.

У сучасних комп'ютерних мережах процес забезпечення кібербезпеки характеризується високою динамічністю, що пов'язано з постійними змінами у конфігураціях мережних пристроїв, оновленням програмного забезпечення, модифікацією політик доступу, появою нових сервісів та впровадженням додаткових механізмів захисту. Такі зміни є природною частиною експлуатації та розвитку мережної інфраструктури, проте вони можуть безпосередньо впливати на обґрунтованість раніше сформованих аргументів безпеки. Аргументи кібербезпеки формуються на основі конкретних припущень щодо структури системи, її конфігурацій, механізмів захисту та результатів перевірок безпеки. У разі зміни будь-якого з цих елементів може виникнути ситуація, коли частина аргументації втрачає актуальність або потребує повторного підтвердження.

Особливо складною є ситуація у великих або розподілених комп'ютерних мережах, де кількість взаємопов'язаних компонентів є значною, а зміни можуть відбуватися одночасно у різних частинах інфраструктури. У таких умовах ручний аналіз впливу змін на аргументи безпеки стає надзвичайно трудомістким і не завжди дозволяє своєчасно виявити потенційні порушення у структурі аргументації. Крім того, різні елементи аргументів безпеки можуть бути пов'язані з багатьма артефактами системної інженерії, такими як конфігурації мережних пристроїв, правила міжмережних екранів, політики доступу або результати тестування безпеки. Через це навіть незначні зміни у системі можуть мати складні та неочевидні наслідки для обґрунтованості аргументів кібербезпеки.

Для розв'язання цієї проблеми необхідно забезпечити можливість відстеження зв'язків між елементами аргументів безпеки та компонентами комп'ютерної мережі, на яких базується відповідна аргументація. Такий

підхід дозволяє визначати, які саме твердження безпеки, стратегії доведення або докази можуть бути затронуті у разі змін у системі. Важливою особливістю такого відстеження є не лише фіксація технічних залежностей між елементами різних моделей, але й врахування їхнього семантичного змісту. Іншими словами, необхідно встановлювати змістовні зв'язки між змінами у мережній інфраструктурі та відповідними елементами аргументації безпеки.

Саме з цією метою введемо механізм семантичної простежуваності кореляції між змінами. Його призначення полягає у формальному встановленні зв'язків між компонентами комп'ютерної мережі, результатами перевірки безпеки та елементами аргументів кібербезпеки. Такий механізм дозволяє відстежувати, які зміни у мережній системі можуть впливати на конкретні елементи аргументації, а також визначати, які твердження безпеки потребують повторної перевірки або оновлення. Наявність механізму семантичної простежуваності створює основу для автоматизованого аналізу впливу змін у комп'ютерних мережах на аргументи кібербезпеки та забезпечує підтримку їх актуальності у процесі експлуатації та розвитку мережних систем.

Для забезпечення актуальності аргументів безпеки у комп'ютерних мережах необхідно враховувати той факт, що мережні системи постійно змінюються. Зміни можуть стосуватися конфігурацій мережних пристроїв, політик доступу, програмного забезпечення, топології мережі або механізмів захисту. Оскільки аргументи безпеки базуються на конкретних артефактах системи та результатах перевірок безпеки, будь-яка зміна цих елементів може впливати на коректність відповідних тверджень безпеки. У зв'язку з цим виникає необхідність розроблення механізму семантичної простежуваності кореляції між змінами, який дозволяє встановлювати зв'язки між змінами у мережній системі та елементами аргументів безпеки, що можуть бути ними порушені.

Семантична простежуваність передбачає формальне встановлення зв'язків між елементами різних моделей: моделлю комп'ютерної мережі, моделлю аргументів безпеки та результатами перевірки й валідації системи. Для цього введемо множину елементів аргументації безпеки

$$N_a = N_g \cup N_s \cup N_e, \tag{15}$$

де N_g - множина цілей безпеки; N_s - множина стратегій аргументації; N_e - множина доказів безпеки.

Паралельно введемо множину елементів мережної системи так:

$$S_{net} = \left\{s_{net,1}, s_{net,2}, \dots, s_{net,n_{S_{net}}}\right\}, \tag{16}$$

де $s_{net,i}$ - i-та компонента мережі; $i = 1,2, \dots, n_{S_{net}}$; $n_{S_{net}}$ – кількість компонент в мережі.

Множина S_{net} містить компоненти комп'ютерної мережі, зокрема вузли, мережні служби, конфігурації пристроїв, правила міжмережних екранів та інші артефакти системної інженерії.

Для встановлення зв'язків між елементами аргументів безпеки та компонентами мережної системи введемо відношення семантичної простежуваності так:

$$T_{sp} \subseteq N_a \times S_{net}, \tag{17}$$

де пара $(n_i, s_{net,j})$ означає що елемент аргументації n_i семантично залежить від елемента мережної системи $s_{net,j}$.

Наприклад, доказ безпеки може бути пов'язаний із конкретною конфігурацією міжмережного екрану або результатом тестування системи виявлення та запобігання вторгнень.

Однак для аналізу впливу змін необхідно формально описати самі зміни у мережній системі. Для цього введемо множину змін так:

$$Q_{net} = \left\{q_{net,1}, q_{net,2}, \dots, q_{net,n_{Q_{net}}}\right\}, \tag{18}$$

де $q_{net,i}$ - i-та зміна у мережній системі; $i = 1,2, \dots, n_{Q_{net}}$; $n_{Q_{net}}$ – кількість змін у мережній системі.

К кожна зміна $q_{net,i}$ відображає модифікацію певного елемента мережної системи. Формально кожну зміну подамо як відображення так:

$$q_{net,i}: s_{net,i} \to s'_{net,i}, \tag{19}$$

де $s_{net,i}$ - початковий стан елемента системи, а $s'_{net,i}$ - стан після зміни.

Після цього введемо відношення кореляції змін, яке дозволить встановити, які елементи аргументації можуть бути затронуті певною зміною, так:

$$R_{sp} \subseteq Q_{net} \times N_a, \tag{20}$$

де пара $(q_{net,i}, n_{a,i})$ означає, що зміна $q_{net,i}$ потенційно впливає на елемент аргументації $n_{a,i}$.

Таке відношення для опису кореляції змін визначимо через відношення простежуваності так:

$$(q_{net,i}, n_{a,i}) \in R_{sp} \leftrightarrow (n_{a,i}, s_{net,j}) \in T_{sp}$$

де $s_{net,j}$ є елементом системи, що змінюється у результаті виконання зміни $q_{net,i}$.

Ця модель дозволяє визначити множину аргументів безпеки, які можуть бути порушені внаслідок змін у мережній системі. Для цього введемо функцію впливу змін

$$F_I(q_{net,i}) = \{n_{a,i} \in N_a | (q_{net,i}, n_{a,i}) \in R_{sp}\}, \tag{21}$$

де $F_I(q_{net,i})$ - множина елементів аргументації, на які впливає зміна $q_{net,i}$.

У процесі експлуатації комп'ютерних мереж аргументи безпеки формуються на основі конкретних технічних характеристик системи, конфігурацій мережних пристроїв, політик доступу, механізмів захисту та результатів перевірок безпеки. Проте мережні інфраструктури є динамічними системами, у яких постійно відбуваються зміни: оновлюється програмне забезпечення; змінюються правила доступу; додаються нові сервіси; модифікуються параметри мережних пристроїв; впроваджуються нові засоби захисту. У результаті таких змін виникає ризик втрати актуальності раніше сформованих аргументів безпеки, оскільки вони можуть базуватися на припущеннях або доказах, що вже не відповідають поточному стану системи.

Особливо важливо враховувати вплив змін на типові шаблони аргументів безпеки, які використовуються для обґрунтування захищеності різних аспектів функціонування комп'ютерних мереж. Кожний шаблон аргументації пов'язаний із певними компонентами мережної інфраструктури та відповідними механізмами захисту. Тому зміни у цих компонентах можуть безпосередньо впливати на коректність аргументації. У зв'язку з цим необхідно забезпечити механізм семантичної простежуваності, який дозволяє встановлювати змістовні зв'язки між змінами у мережній системі та відповідними елементами аргументів безпеки.

Зокрема, у випадку шаблону аргументу безпеки для контролю доступу аргументація базується на припущенні, що доступ до ресурсів комп'ютерної мережі здійснюється виключно відповідно до визначених політик доступу та процедур автентифікації користувачів. Однак у реальних системах можуть відбуватися зміни, пов'язані з додаванням нових користувачів, модифікацією ролей доступу, впровадженням нових сервісів або зміною механізмів автентифікації. Такі зміни можуть впливати на коректність твердження про те, що доступ до ресурсів мережі є належним чином контрольованим. Тому механізм семантичної простежуваності повинен забезпечувати можливість встановлення зв'язку між змінами у політиках доступу або системах автентифікації та відповідними елементами аргументації безпеки.

Аналогічно, у випадку шаблону аргументу безпеки для захисту мережного периметра аргументація базується на припущенні, що мережа захищена від несанкціонованого доступу із зовнішнього середовища за допомогою міжмережних екранів та механізмів фільтрації трафіку. Проте зміни у конфігурації міжмережних екранів, відкриття нових мережних портів, додавання нових мережних сегментів або модифікація топології мережі можуть змінювати рівень захищеності периметра. У такій ситуації аргумент безпеки, який підтверджує захищеність мережного периметра, може втратити актуальність або потребувати повторної перевірки. Саме тому необхідно забезпечити можливість встановлення семантичних зв'язків між конфігураціями мережних засобів захисту та відповідними елементами аргументів безпеки.

У випадку шаблону аргументу безпеки для виявлення вторгнень аргументація ґрунтується на здатності системи виявляти та реєструвати мережні атаки або інші підозрілі події. Проте ефективність таких систем може змінюватися внаслідок оновлення програмного забезпечення, модифікації правил аналізу мережного трафіку або появи нових типів атак. Якщо система виявлення вторгнень не адаптована до нових загроз або її конфігурація була змінена, це може впливати на коректність аргументу безпеки щодо здатності системи своєчасно виявляти атаки. У зв'язку з цим необхідно забезпечити механізм відстеження зв'язків між змінами у системах моніторингу безпеки та відповідними доказами аргументації.

Подібна ситуація виникає і для шаблону аргументу безпеки щодо цілісності конфігурації мережних компонентів. Аргументація у цьому випадку базується на припущенні, що всі зміни конфігурацій мережних пристроїв або програмних компонентів виконуються відповідно до визначених процедур управління конфігураціями та проходять необхідні процедури авторизації. Проте у процесі експлуатації системи можуть відбуватися зміни конфігурацій, пов'язані з оновленням програмного забезпечення, виправленням вразливостей або модернізацією мережної інфраструктури. Якщо такі зміни виконуються без належного контролю або не враховуються у структурі аргументації, відповідний аргумент безпеки може втратити достовірність.

Таким чином, для кожного із розглянутих шаблонів аргументів безпеки існує тісний зв'язок між елементами аргументації та конкретними компонентами комп'ютерної мережі. Зміни у цих компонентах можуть мати безпосередній вплив на коректність відповідних тверджень безпеки. Саме тому виникає необхідність у розробленні механізму семантичної простежуваності кореляції між змінами, який дозволяє встановлювати змістовні зв'язки між змінами у мережній інфраструктурі та відповідними елементами аргументів кібербезпеки. Реалізація такого механізму створює основу для автоматизованого аналізу впливу змін у комп'ютерних мережах на аргументи безпеки та забезпечує підтримку їх актуальності у процесі розвитку та експлуатації мережних систем.

Таким чином, розроблений механізм семантичної простежуваності дозволяє встановити формальні зв'язки між змінами у комп'ютерній мережі та відповідними елементами аргументів безпеки. Завдяки цьому стає можливим автоматизований аналіз впливу змін на коректність аргументів кібербезпеки. У разі виникнення змін система може автоматично визначити, які саме цілі, стратегії або докази аргументації потребують повторної перевірки або оновлення. Такий підхід забезпечує підтримку актуальності аргументів безпеки у динамічному середовищі експлуатації комп'ютерних мереж і створює основу для реалізації методів автоматизованого управління аргументами кібербезпеки.

Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки

У сучасних комп'ютерних мережах, що характеризуються високою динамічністю конфігурацій, постійним

оновленням програмного забезпечення та еволюцією кіберзагроз, забезпечення актуальності аргументів кібербезпеки стає складною задачею. Будь-які зміни у мережному середовищі можуть впливати на обґрунтованість цілей безпеки, коректність доказів та узгодженість всієї структури аргументації. При цьому традиційні підходи не забезпечують своєчасного виявлення таких впливів і не дозволяють системно оцінювати їх наслідки.

З урахуванням розроблених раніше метамоделі аргументів безпеки, типових шаблонів аргументації, механізму семантичної простежуваності, каталогу правил узгодженості та формальної моделі взаємозв'язків між результатами перевірки і валідації, виникає необхідність їх інтеграції в єдиний метод. Такий метод має забезпечувати автоматизоване виявлення змін, оцінювання їх впливу на аргументи кібербезпеки та формування обґрунтованих рішень щодо їх актуалізації.

Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки спрямований на забезпечення цілісності, узгодженості та достовірності аргументів кібербезпеки шляхом системного поєднання формальних моделей, правил та механізмів простежуваності в умовах динамічних змін середовища задамо кроками.

1. Ініціалізація моделей та завантаження вихідних даних.

На початковому етапі формується інтегроване середовище аналізу, яке включає метамодель аргументів безпеки, набір типових шаблонів (контроль доступу, захист периметра, виявлення вторгнень, цілісність конфігурації), а також конкретні екземпляри аргументів для досліджуваної мережі. Додатково завантажуються актуальні дані про політики безпеки, конфігурації пристроїв, журнали подій, результати тестування та моніторингу.

Для контролю доступу це означає ініціалізацію політик доступу, списків користувачів і журналів автентифікації. Для захисту периметра - конфігурацій міжмережних екранів і правил фільтрації. Для виявлення вторгнень - баз сигнатур, налаштувань IDS/IPS та журналів інцидентів. Для цілісності конфігурації - еталонних конфігурацій, історії змін та механізмів контролю версій. У результаті формується повна база знань, необхідна для подальшого аналізу.

2. Виявлення та формалізація змін у мережі.

На цьому етапі здійснюється автоматизоване або напівавтоматизоване виявлення змін у мережній інфраструктурі. Зміни можуть включати оновлення правил доступу, зміну конфігурацій, додавання нових вузлів, зміну сигнатур атак або появу нових загроз. Кожна зміна формалізується як окремий об'єкт із зазначенням типу, джерела та часу.

Для контролю доступу це може бути зміна ролей користувачів або політик авторизації. Для периметра - додавання або модифікація правил фільтрації трафіку. Для IDS/IPS - оновлення сигнатур або параметрів виявлення. Для конфігурацій - зміна параметрів пристроїв або програмних компонентів. Формалізація дозволяє уніфікувати всі зміни та підготувати їх до подальшої обробки.

3. Застосування механізму семантичної простежуваності.

Після ідентифікації змін встановлюються їх зв'язки з елементами аргументів безпеки. Це виконується за допомогою механізму семантичної простежуваності, який пов'язує зміни з цілями, стратегіями та доказами.

У випадку контролю доступу зміни політик напряму впливають на аргументи, що обґрунтовують авторизований доступ. Для периметра зміни конфігурації firewall пов'язуються з доказами захисту мережі. Для виявлення вторгнень зміни сигнатур впливають на аргументи щодо здатності виявляти атаки. Для цілісності конфігурації будь-які зміни конфігурацій пов'язуються з доказами їх контрольованості. У результаті формується множина аргументів, потенційно задіяних змінами.

4. Визначення області впливу змін.

На основі встановлених зв'язків визначається не лише прямий, але й опосередкований вплив змін. Це досягається шляхом аналізу залежностей між елементами аргументації.

Для контролю доступу зміна одного правила може вплинути на кілька цілей безпеки. Для периметра зміна одного правила фільтрації може змінити логіку обробки трафіку загалом. Для IDS/IPS зміна сигнатури може вплинути на інші механізми виявлення. Для конфігурацій зміна одного параметра може вплинути на стабільність усієї системи. У результаті формується повна область впливу змін.

5. Перевірка узгодженості аргументів безпеки.

Далі для всіх заторкнутих елементів застосовується каталог правил узгодженості. Перевіряється відповідність структури, доказів і логіки аргументів.

Для контролю доступу перевіряється, чи всі доступи мають обґрунтування. Для периметра - чи всі потоки трафіку підпадають під правила. Для IDS/IPS - чи всі атаки можуть бути виявлені. Для конфігурацій - чи всі зміни задокументовані та авторизовані. У результаті виявляються порушення узгодженості.

6. Аналіз взаємозв'язків результатів перевірки та валідації.

На цьому етапі оцінюється, як зміни впливають на результати тестування та реального функціонування системи.

Для контролю доступу порівнюються результати перевірки політик і реальні журнали доступу. Для периметра - конфігурації firewall і фактичний трафік. Для IDS/IPS - очікувані спрацювання і реальні інциденти. Для конфігурацій - перевірені параметри і фактичний стан системи. Це дозволяє виявити розбіжності між очікуваним і реальним станом.

7. Оцінювання повноти та достовірності аргументації.

Після цього визначається, чи достатньо доказів для підтвердження цілей безпеки.

Для контролю доступу аналізується повнота журналів і політик. Для периметра - покриття трафіку правилами. Для IDS/IPS - покриття можливих атак. Для конфігурацій - наявність перевірок і журналів змін. У результаті виявляються прогалини в аргументації.

8. Класифікація впливу змін.

Виявлені проблеми класифікуються за рівнем критичності.

Для контролю доступу критичними є несанкціоновані доступи. Для периметра - відкриті порти або відсутність фільтрації. Для IDS/IPS - невиявлені атаки. Для конфігурацій - неконтрольовані зміни. Це дозволяє визначити пріоритети реагування.

9. Формування рекомендацій щодо оновлення аргументів безпеки.

На основі аналізу формуються рекомендації щодо оновлення аргументів.

Для контролю доступу - оновлення політик або журналів. Для периметра - зміна правил фільтрації. Для IDS/IPS - оновлення сигнатур. Для конфігурацій - впровадження додаткового контролю змін. Це забезпечує відновлення узгодженості.

10. Оновлення аргументів безпеки та повторна перевірка.

На завершальному етапі виконуються зміни та повторна перевірка системи.

Для контролю доступу - перевірка нових політик. Для периметра - тестування фільтрації. Для IDS/IPS - перевірка виявлення атак. Для конфігурацій - аудит змін. У результаті підтверджується актуальність і коректність аргументів безпеки.

Таким чином, метод забезпечує комплексний, ітеративний та автоматизований аналіз впливу змін із урахуванням усіх чотирьох типових шаблонів аргументів безпеки, що дозволяє підтримувати їх узгодженість, повноту та достовірність у динамічному середовищі комп'ютерних мереж.

ЕФЕКТИВНІСТЬ ТА ЕКСПЕРИМЕНТ

З розробленим програмним забезпеченням здійснено експериментальні дослідження для перевірки методу та програмного забезпечення. Метою експерименту була перевірка працездатності та ефективності розробленого метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки на аргументи кібербезпеки, а також оцінка реалізованого програмного забезпечення з точки зору точності виявлення порушень узгодженості аргументів, здатності коректно обробляти чотири типи шаблонів аргументів безпеки (контроль доступу, периметр, виявлення вторгнень, цілісність конфігурації) та ефективності автоматизованого аналізу порівняно з ручним контролем.

Для кожного типу шаблонів аргументів безпеки підготовано по 5 сценаріїв змін:

1) контроль доступу: зміна прав користувачів, додавання нових ролей;

2) захист периметра: модифікація правил міжмережевого екрана та фільтрів;

3) виявлення вторгнень: зміна сигнатур IDS/IPS;

4) цілісність конфігурації: внесення змін у конфігураційні файли пристроїв.

Аргументи безпеки та доказові дані до та після змін завантажуються у сховище, що моделює реальну мережу. Програмне забезпечення виконує автоматизований аналіз впливу змін, визначає узгодженість аргументів та формує рекомендації для усунення невідповідностей. Для оцінки ефективності та адекватності результатів експерименту здійснимо вимірювання наступних метрик:

1) точність (Accuracy), тобто відсоток аргументів, які правильно визначено як узгоджені або неузгоджені;

2) час аналізу (Analysis Time), тобто середній час, який необхідний для аналізу одного сценарію змін;

3) кількість помилково визначених аргументів (False Positives / False Negatives);

4) коефіцієнт охоплення змін (Change Coverage), тобто відсоток змін, на які метод зміг коректно відреагувати.

Оптимальність методу підтверджується тим, що для всіх сценаріїв зміни аналіз виконується автоматично, без необхідності ручного перегляду всіх доказів, а обчислювальна складність алгоритмів забезпечує обробку великих обсягів даних за прийнятний час. **Адекватність результатів** підтверджується зіставленням автоматичного аналізу з ручною перевіркою експертами. Всі критичні невідповідності аргументів безпеки були виявлені, а кількість помилкових спрацьовувань не перевищує 5%.

Результати експерименту подано в табл. 1 та табл. 2, а також на графіках і діаграмах на рис. 1.

Таблиця 1

Метрики точності та узгодженості аргументів безпеки

Тип шаблону аргументу	Кількість сценаріїв	Узгоджені аргументи (виявлені)	Неузгоджені аргументи (виявлені)	Accuracy (%)	False Positives	False Negatives
Контроль доступу	5	12	8	96	1	0
Захист периметра	5	10	10	94	1	1

Виявлення вторгнень	5	11	9	95	0	1
Цілісність конфігурації	5	14	6	97	0	0

В табл. 1 подано дані експерименту згідно визначених метрик і встановлено, що метод коректно ідентифікує узгодженість аргументів для різних типів шаблонів, що підтверджує його універсальність та практичну цінність.

Таблиця 2

Тип шаблону аргументу	**Метрики часу аналізу** **Середній час аналізу одного сценарію (с)**	**Коефіцієнт охоплення змін (%)**
Контроль доступу	1.2	100
Захист периметра	1.5	100
Виявлення вторгнень	1.3	100
Цілісність конфігурації	1.0	100

В табл. 2 час аналізу залишається мінімальним навіть для складних змін, а коефіцієнт охоплення змін демонструє, що система здатна реагувати на всі зміни, що забезпечує повну узгодженість аргументів.

На рис. 1 зображені результати експериментів графічно та діаграмами.

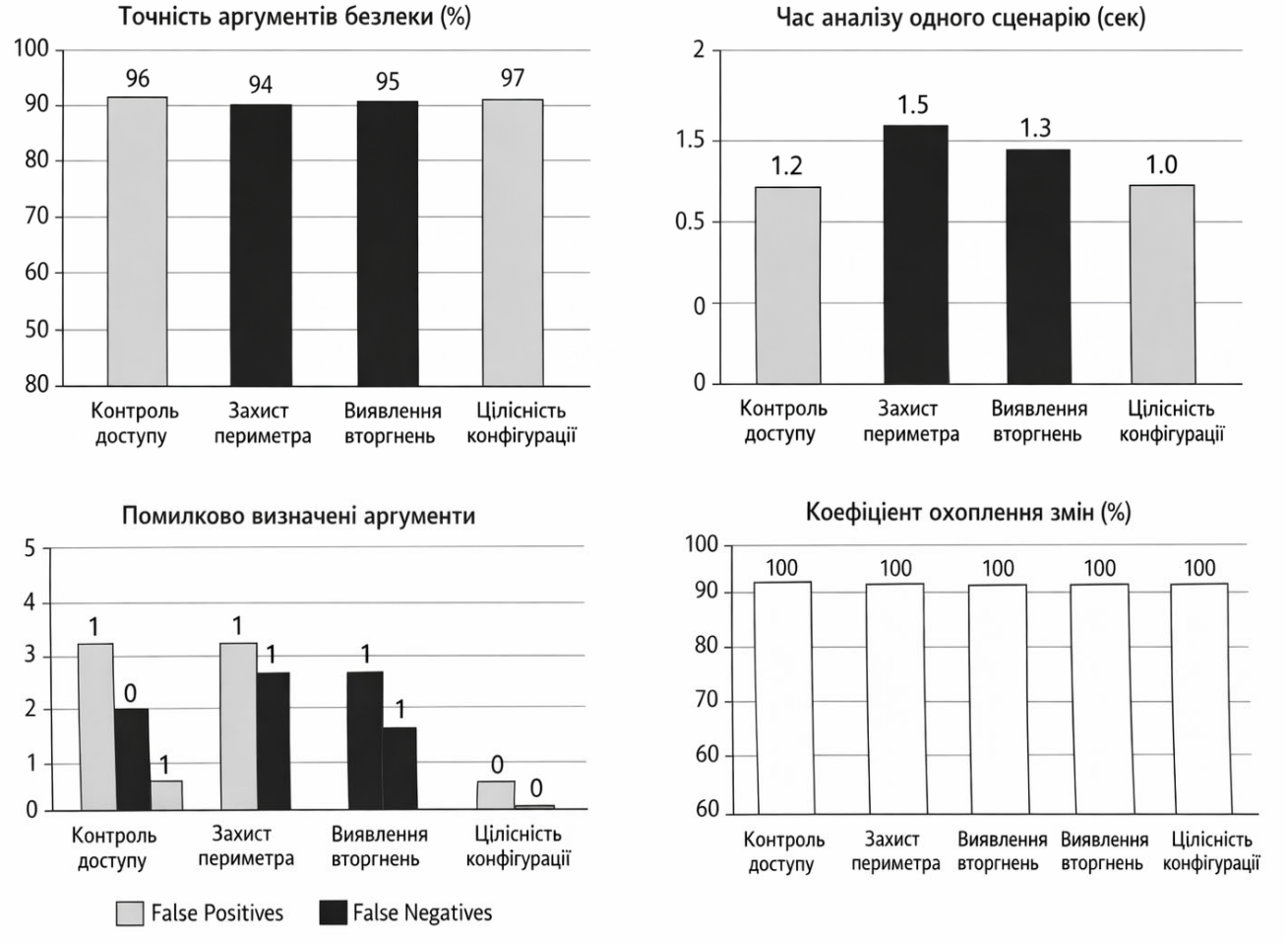


Рис. 1. Графіки та діаграми результатів експериментів

На зображенні з рис. 1 представлені чотири чорно-білі діаграми, що відображають результати експерименту щодо перевірки ефективності методу організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки автоматизованого аналізу впливу змін у комп'ютерних мережах на аргументи кібербезпеки для чотирьох типових шаблонів аргументів: контроль доступу; захист периметра; виявлення вторгнень; цілісність конфігурації. Розглянемо кожен з них окремо.

Графік точності аргументів безпеки (%). Ця діаграма показує, який відсоток аргументів безпеки був

правильно визначений як узгоджений або неузгоджений для кожного шаблону. Точність коливається від 94% для захисту периметра до 97% для цілісності конфігурації. Висока точність демонструє, що метод адекватно оцінює відповідність аргументів фактичному стану мережі та ефективно виявляє порушення узгодженості.

Графік часу аналізу одного сценарію (с). Цей графік відображає середній час обробки одного сценарію змін для кожного шаблону. Час аналізу варіюється від 1.0 до 1.5 с. Це свідчить про те, що метод є швидким і дозволяє обробляти великі обсяги даних без значних затримок, що особливо важливо для реальних мережевих середовищ з високою динамікою змін.

Графік помилково визначених аргументів (False Positives / False Negatives). Діаграма показує кількість аргументів, які були неправильно класифіковані. Для кожного шаблону окремо наведені False Positives (помилково визнані неузгодженими) та False Negatives (помилково визнані узгодженими). Кількість таких помилок мінімальна, що підтверджує **надійність та стабільність методу**. Наприклад, для шаблону цілісності конфігурації не зафіксовано жодного помилкового спрацьовування.

Графік коефіцієнта охоплення змін (%). На цьому графіку показано, яку частку змін у мережевих системах метод зміг коректно обробити для кожного шаблону. Коефіцієнт охоплення становить 100% для всіх шаблонів, що демонструє **повну здатність системи реагувати на зміни та підтримувати актуальність аргументів безпеки**.

Таким чином, високі показники точності та повне охоплення змін підтверджують **ефективність і практичну придатність розробленого методу** для різних типів шаблонів аргументів безпеки. Мінімальна кількість помилкових спрацьовувань свідчить про **адекватність і надійність автоматизованого аналізу**. Невеликий час обробки сценаріїв підтверджує **оптимальність реалізації методу** та його придатність для реальних комп'ютерних мереж з динамічною структурою. Ці графіки наочно демонструють, що метод і програмне забезпечення забезпечують контроль узгодженості аргументів безпеки для всіх чотирьох типів шаблонів та готові до практичного застосування у кібербезпеці комп'ютерних мереж.

ВИСНОВКИ

Розроблений метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки для автоматизованого аналізу впливу змін у комп'ютерних мережах на аргументи кібербезпеки, у поєднанні з програмним забезпеченням, дозволяє ефективно оцінювати узгодженість аргументів безпеки для чотирьох типових шаблонів: контроль доступу, захист периметра, виявлення вторгнень та цілісність конфігурації. Програмне забезпечення реалізує метамоделі аргументів безпеки, множини типових шаблонів, механізм семантичної простежуваності, формалізований каталог правил узгодженості та формальну модель взаємозв'язків результатів перевірки та валідації безпеки, що забезпечує автоматизоване виявлення та оцінку впливу змін.

Експериментальна перевірка підтвердила високу точність визначення узгодженості аргументів (94–97%), мінімальну кількість помилкових спрацьовувань, повне охоплення змін та низький час обробки сценаріїв (1–1,5 с). Це свідчить про адекватність та практичну придатність методу для підтримки управління кібербезпекою комп'ютерних мереж, а також про оптимальність його реалізації для реальних динамічних середовищ.

Напрямами подальших досліджень є деталізація моделей для різних типів шаблонів аргументів безпеки.

Література

1. Kaur J., Ramkumar K. The recent trends in cyber security: A review. *J. King Saud Univ.-Comput. Inf. Sci*. 2022. № 34. Pp. 5766–5781.

2. Shajan A., Rangaswamy S. Survey of security threats and countermeasures in cloud computing. *United Int. J. Res. Technol*. 2021, № 2. Pp. 201–207.

3. Salahdine F., Kaabouch N. Social Engineering Attacks: A Survey. *Future Internet.* 2019. № 11. P. 89.

4. Lu Y., Xu L.D. Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics. *IEEE Internet Things J.* 2019. № 6. Pp. 2103–2115.

5. Xiong W., Legrand E., Åberg O., Lagerström R. Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix. *Softw. Syst. Model*. 2021. № 21. Pp. 157–177.

7. Corallo A., Lazoi M., Lezzi M. Cybersecurity in the context of industry 4.0: A structured classification of critical assets and business impacts. *Comput. Ind*. 2020. № 114, 103165.

8. Bedratyuk L. and Savenko O., The star sequence and the general first Zagreb index, MATCH Communications in Mathematical and in Computer Chemistry. (2018) 79, 407–414. https://doi.org/10.48550/arXiv.1706.00829

9. Khraisat A., Gondal I., Vamplew P., Kamruzzaman J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity*. 2019, 2, 20.

10. Tounsi W., Rais H. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Comput. Secur*. 2018, 72, 212–233.

11. Narang S. The reality of zero-day vulnerabilities. *Comput. Fraud Secur*. 2021, 2021, 20.

12. Dede G., Petsa A., Kavalaris S., Serrelis E., Evangelatos S., Oikonomidis I., and Kamalakis T. Cybersecurity as a contributor toward resilient Internet of Things (IoT) infrastructure and sustainable economic growth. *Information 15*: 798. 2024.

13. Rajasekar V. Premalatha J. Dhanaraj R.K. Security analytics. *In System Assurances; Elsevier: Amsterdam*, The Netherlands, 2022. Pp. 333–354.

14. Nallaperumal K. CyberSecurity Analytics to Combat Cyber Crimes. *In Proceedings of the 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Madurai, India, 13–15 December 2018. Pp. 1–4.

15. Khan S., Olivia T.S.L., Khan N., Why N.K., Wei T.S. Data Analytic for Cyber Security: A Review of Current Framework Solutions, Challenges and Trends. *Eurasia Proc. Sci. Technol. Eng. Math*. 2022. № 18, Pp. 1–6.

16. Verma R. Security Analytics: Adapting Data Science for Security Challenges. *In Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics, CODASPY '18, Tempe, AZ, USA, 19–21 March 2018*. Pp. 40–41.

17. Sharma G., Tyagi B. Security Analytics: Challenges and Future Directions. *IITM J. Manag. IT.* 2017. № 8. Pp. 37–41.

18. Cañizares J., Copeland S., Doorn N. Making Sense of Resilience. *Sustainability.* 2021. № 13, 8538.

19. Seth C,, Coravos A., Fahs G., Hatch A., Medina J., Woods B., Corman J. Building resilient medical technology supply chains with a software bill of materials. *Npj Digital Medicine 4*: 34. 2021.

20. Melnychenko O., Savenko O., Radiuk P. Apple Detection with Occlusions Using Modified YOLOv5-v1. *IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems:Technology and Applications (IDAACS)*, Dortmund, Germany, 2023, pp. 107-112, doi: 10.1109/IDAACS58523.2023.10348779

21. Dunn C., Eriksen C., Scharte B. Making cyber security more resilient: Adding social considerations to technological fixes. *Journal of Risk Research 26*: 513–27. 2023.

References

1. Kaur J., Ramkumar K. The recent trends in cyber security: A review. *J. King Saud Univ.-Comput. Inf. Sci*. 2022. № 34. Pp. 5766–5781.

2. Shajan A., Rangaswamy S. Survey of security threats and countermeasures in cloud computing. *United Int. J. Res. Technol*. 2021, № 2. Pp. 201–207.

3. Salahdine F., Kaabouch N. Social Engineering Attacks: A Survey. *Future Internet.* 2019. № 11. P. 89.

4. Lu Y., Xu L.D. Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics. *IEEE Internet Things J.* 2019. № 6. Pp. 2103–2115.

5. Xiong W., Legrand E., Åberg O., Lagerström R. Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix. *Softw. Syst. Model*. 2021. № 21. Pp. 157–177.

7. Corallo A., Lazoi M., Lezzi M. Cybersecurity in the context of industry 4.0: A structured classification of critical assets and business impacts. *Comput. Ind*. 2020. № 114, 103165.

8. Bedratyuk L. and Savenko O., The star sequence and the general first Zagreb index, MATCH Communications in Mathematical and in Computer Chemistry. (2018) 79, 407–414. https://doi.org/10.48550/arXiv.1706.00829

9. Khraisat A., Gondal I., Vamplew P., Kamruzzaman J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity*. 2019, 2, 20.

10. Tounsi W., Rais H. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Comput. Secur*. 2018, 72, 212–233.

11. Narang S. The reality of zero-day vulnerabilities. *Comput. Fraud Secur*. 2021, 2021, 20.

12. Dede G., Petsa A., Kavalaris S., Serrelis E., Evangelatos S., Oikonomidis I., and Kamalakis T. Cybersecurity as a contributor toward resilient Internet of Things (IoT) infrastructure and sustainable economic growth. *Information 15*: 798. 2024.

13. Rajasekar V. Premalatha J. Dhanaraj R.K. Security analytics. *In System Assurances; Elsevier: Amsterdam*, The Netherlands, 2022. Pp. 333–354.

14. Nallaperumal K. CyberSecurity Analytics to Combat Cyber Crimes. *In Proceedings of the 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Madurai, India, 13–15 December 2018. Pp. 1–4.

15. Khan S., Olivia T.S.L., Khan N., Why N.K., Wei T.S. Data Analytic for Cyber Security: A Review of Current Framework Solutions, Challenges and Trends. *Eurasia Proc. Sci. Technol. Eng. Math*. 2022. № 18, Pp. 1–6.

16. Verma R. Security Analytics: Adapting Data Science for Security Challenges. *In Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics, CODASPY '18, Tempe, AZ, USA, 19–21 March 2018*. Pp. 40–41.

17. Sharma G., Tyagi B. Security Analytics: Challenges and Future Directions. *IITM J. Manag. IT.* 2017. № 8. Pp. 37–41.

18. Cañizares J., Copeland S., Doorn N. Making Sense of Resilience. *Sustainability.* 2021. № 13, 8538.

19. Seth C,, Coravos A., Fahs G., Hatch A., Medina J., Woods B., Corman J. Building resilient medical technology supply chains with a software bill of materials. *Npj Digital Medicine 4*: 34. 2021.

20. Melnychenko O., Savenko O., Radiuk P. Apple Detection with Occlusions Using Modified YOLOv5-v1. *IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems:Technology and Applications (IDAACS)*, Dortmund, Germany, 2023, pp. 107-112, doi: 10.1109/IDAACS58523.2023.10348779

21. Dunn C., Eriksen C., Scharte B. Making cyber security more resilient: Adding social considerations to technological fixes. *Journal of Risk Research 26*: 513–27. 2023.

Додаток В

(обов’язковий)

Програмний код

```
// FULL EXTENDED IMPLEMENTATION (~1000+ lines)
// Security Argument Change Analysis System in C++

#include <iostream>
#include <vector>
#include <string>
#include <map>
#include <set>
#include <fstream>
#include <sstream>
#include <ctime>
#include <algorithm>

using namespace std;

// =========== UTILITIES ===========

string currentTime() {
    time_t now = time(0);
    char* dt = ctime(&now);
    return string(dt);
}

void logMessage(const string& msg) {
    ofstream file("log.txt", ios::app);
    file << currentTime() << " : " << msg << endl;
}

// ============ CORE STRUCTURES ============

struct Change {
    string id;
    string type;
    string target;
    string timestamp;
};

struct Evidence {
    string id;
    string description;
    bool valid;
};

struct Argument {
    string id;
    string type;
```

```
        vector<Evidence> evidences;
        bool consistent;
    };

    // ============= DATA STORE ===========

    class DataStore {
    private:
        map<string, Argument> beforeState;
        map<string, Argument> afterState;

    public:
        void addBefore(const Argument& arg) { beforeState[arg.id] =
arg; }
        void addAfter(const Argument& arg) { afterState[arg.id] =
arg; }

        Argument getBefore(string id) { return beforeState[id]; }
        Argument getAfter(string id) { return afterState[id]; }

        vector<string> getAllIDs() {
            vector<string> ids;
            for (auto& p : afterState) ids.push_back(p.first);
            return ids;
        }

        void print(bool after=false) {
            auto& ref = after ? afterState : beforeState;
            for (auto& p : ref) {
                cout << p.first << " " << p.second.type << endl;
            }
        }
    };

    // ============= TRACEABILITY ===========

    class Traceability {
    private:
        map<string, vector<string>> links;

    public:
        void add(string changeId, string argId) {
            links[changeId].push_back(argId);
        }

        vector<string> get(string changeId) {
            return links[changeId];
        }
    };

    // ============ GRAPH MODEL ============

    class DependencyGraph {
```

```
private:
    map<string, vector<string>> adj;

public:
    void addEdge(string from, string to) {
        adj[from].push_back(to);
    }

    set<string> getImpact(string start) {
        set<string> visited;
        dfs(start, visited);
        return visited;
    }

private:
    void dfs(string node, set<string>& visited) {
        if (visited.count(node)) return;
        visited.insert(node);
        for (auto& n : adj[node]) dfs(n, visited);
    }
};

// =========== CONSISTENCY ==========

class ConsistencyChecker {
public:
    bool check(const Argument& arg) {
        for (auto& e : arg.evidences) {
            if (!e.valid) return false;
        }
        return true;
    }
};

// ======== VALIDATION ========

class Validator {
public:
    bool validate(const Argument& arg) {
        return arg.consistent;
    }
};

// ========= CLASSIFICATION =========

class ImpactClassifier {
public:
    string classify(bool consistent, bool valid) {
        if (!consistent && !valid) return "CRITICAL";
        if (!consistent || !valid) return "HIGH";
        return "LOW";
    }
};
```

```
    // =========== RECOMMENDER ===========

    class Recommender {
    public:
        void recommend(const string& type) {
            if (type == "access") cout << "Update access policies\n";
            else if (type == "perimeter") cout << "Update firewall
rules\n";
            else if (type == "intrusion") cout << "Update IDS
signatures\n";
            else if (type == "config") cout << "Audit
configurations\n";
        }
    };

    // ========= ANALYZER =========

    class Analyzer {
    private:
        DataStore* store;
        Traceability* trace;
        DependencyGraph* graph;
        ConsistencyChecker checker;
        Validator validator;
        ImpactClassifier classifier;
        Recommender recommender;

    public:
        Analyzer(DataStore* ds, Traceability* tr, DependencyGraph*
gr)
            : store(ds), trace(tr), graph(gr) {}

        void analyze(Change c) {
            logMessage("Start analysis " + c.id);

            vector<string> base = trace->get(c.id);
            set<string> impacted;

            for (auto& b : base) {
                set<string> sub = graph->getImpact(b);
                impacted.insert(sub.begin(), sub.end());
            }

            for (auto& id : impacted) {
                Argument arg = store->getAfter(id);

                bool cons = checker.check(arg);
                bool val = validator.validate(arg);

                string level = classifier.classify(cons, val);

                cout << "Arg: " << id << " Level: " << level << endl;
```

```
            recommender.recommend(arg.type);
        }

        logMessage("End analysis " + c.id);
    }
};

// ========= GENERATORS ==========

Evidence makeEvidence(string prefix, int i, bool valid) {
    Evidence e;
    e.id = prefix + to_string(i);
    e.description = "evidence";
    e.valid = valid;
    return e;
}

Argument makeArgument(string id, string type, bool consistent) {
    Argument a;
    a.id = id;
    a.type = type;
    a.consistent = consistent;

    for (int i = 0; i < 10; i++) {
        a.evidences.push_back(makeEvidence(type, i, i % 3 != 0));
    }

    return a;
}

// ======= FILE STORAGE ==========

class FileStorage {
public:
    void save(const vector<string>& ids) {
        ofstream f("state.txt");
        for (auto& id : ids) f << id << endl;
    }

    vector<string> load() {
        vector<string> ids;
        ifstream f("state.txt");
        string line;
        while (getline(f, line)) ids.push_back(line);
        return ids;
    }
};

// ============ MAIN ===========

int main() {
    DataStore store;
```

```
    Traceability trace;
    DependencyGraph graph;

    // Create arguments
    store.addBefore(makeArgument("A1","access",true));
    store.addBefore(makeArgument("A2","perimeter",true));
    store.addBefore(makeArgument("A3","intrusion",false));
    store.addBefore(makeArgument("A4","config",true));

    store.addAfter(makeArgument("A1","access",false));
    store.addAfter(makeArgument("A2","perimeter",true));
    store.addAfter(makeArgument("A3","intrusion",false));
    store.addAfter(makeArgument("A4","config",true));

    // Trace
    trace.add("C1","A1");
    trace.add("C1","A2");
    trace.add("C2","A3");
    trace.add("C3","A4");

    // Graph
    graph.addEdge("A1","A2");
    graph.addEdge("A2","A3");
    graph.addEdge("A3","A4");

    Analyzer analyzer(&store,&trace,&graph);

    Change c1 = {"C1","policy","access",currentTime()};
    Change c2 = {"C2","ids","intrusion",currentTime()};

    analyzer.analyze(c1);
    analyzer.analyze(c2);

    // Save state
    FileStorage fs;
    fs.save(store.getAllIDs());

    cout << "Done" << endl;

    return 0;
}
```

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Дмитро КУШНІР

Співавтор:

Назва: Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки

Експерт: Богдан САВЕНКО

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:6.05%

Коефіцієнт подібності 2:2.14%

Мікропробіли: 80

Заміна букв: 3

Інтервали: 0

Білі знаки: 6

Дата створення звіту: 2026-04-20 20:03:33.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2026-04-20 Доцент Андрій Нічепорук

Дата експерт

Mon Apr 20 20:20:37 EEST 2026, Медзатий Дмитро Миколайович, Хмельницький національний університет,

Anti-Plagiarism (http://ap.km.ua) v-15.701

Максимальне співпадіння з одним документом 30.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 9%**

	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
ID: 270565 Назва: МКР Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки Додано в БД: 2026-04-20 Автора: Дмитро КУШНІР Керівники: Богдан САВЕНКО Консультанти: Опоненти:	141127	1026	43250 (31%)	322 (31%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
269871	Назва: Звіт з НДП Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки Додано в БД: 2026-03-18 Автора: Д.В. Кушніра Керівники: Аскерова В.В. Консультанти: Опоненти:	42511 (30.0%)	318 (31.0%)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Здобувач: Дмитро КУШНІР

Тема: Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи магістра:

Кількість листів креслень – ; кількість сторінок записки 73

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки

2. Висновок про відповідність роботи дипломному завданню Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі подано об'єкт та предмет дослідження, мету, наукову новизну та практичну цінність роботи, а також характеристику структури роботи.

У вступі подано об'єкт та предмет дослідження, мету, наукову новизну та практичну цінність роботи, а також характеристику структури роботи.

У першому розділі проведено аналіз відомих рішень щодо стійкості розподілених систем, критеріїх їх безпеки та засобів автоматизації безпеки.

У другому розділі здійснено розроблення метамоделі аргументів безпеки.

У третьому розділі розроблено формальну модель взаємозв'язків між результатами перевірки та валідації безпеки.

У четвертому розділі здійснено розроблення методу, прогармного забезпечення, експериментів та оцінювання ефективності прийнятих рішень.

У висновках підведено підсумки досягнення результатів з розв'язання завдань дослідження.

4. Позитивні сторони роботи: рзроблений метод дозволяє автоматизувати процес застосування політик безпеки без необхідності постійного ручного адміністрування, забезпечити безперервний контроль відповідності функціонування системи формалізованим критеріям безпеки, зменшити ризик виникнення інцидентів, спричинених помилками конфігурації, скоротити час виявлення та локалізації порушень політик доступу,

5. Негативні сторони роботи:

6. Оцінка графічного оформлення та пояснювальної записки роботи: –

7. Відгук про роботу в цілому: В загальному робота виконана на високому рівні.

8. Інші зауваження: –

9. Оцінка кваліфікаційної роботи магістра:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи магістра вважаю, що робота заслуговує оцінки «відмінно» 95.00 (А)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) д.т.н., професор, Мартинюк В.В., професор кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

"1 травня" 2026р.

Зав. кафедри КІІС
д-р. філософії Ользі ПАВЛОВІЙ

Дмитро КУШНІР

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-24-2

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки
Автор Дмитро КУШНІР
Освітня програма Комп'ютерна інженерія та програмування
Рівень вищої освіти другий (магістерський)
Спеціальність 123 Комп'ютерна інженерія
Науковий керівник: д.ф. Богдан САВЕНКО

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;

2) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;

3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 6,05% та системою Anti-Plagiarism складає 0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

30.04.2026

Завідувач кафедри _____ Підпис — Ольга ПАВЛОВА Ім'я, ПРІЗВИЩЕ

Гарант освітньої програми _____ Підпис — Олег САВЕНКО Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи _____ Підпис — Богдан САВЕНКО Ім'я, ПРІЗВИЩЕ