Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра інформаційної та кібернетичної безпеки
імені професора Володимира Бурячка

«Допущено до захисту»
Завідувач кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка
кандидат технічних наук, доцент
Складанний П.М.

_____
(підпис)
« ___ » _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття другого (магістерського)
рівня вищої освіти

Спеціальність 125 Кібербезпека та захист інформації

Тема роботи:
ДОСЛІДЖЕННЯ ТА ВИРОБЛЕННЯ РЕКОМЕНДАЦІЙ ЩОДО МОДЕЛЕЙ АВТЕНТИФІКАЦІЇ ТА АВТОРИЗАЦІЇ В РАМКАХ МЕТОДОЛОГІЇ ZERO TRUST

Виконав

студент групи БІКСм-1-25-1.4д

Тімаков Андрій Іванович
(прізвище, ім'я, по батькові)

_____
(підпис)

Науковий керівник
Кандидат технічних наук, доцент
(науковий ступінь, наукове звання)

Складанний Павло Миколайович
(прізвище, ініціали)

_____
(підпис)

Київ – 2025

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра інформаційної та кібернетичної безпеки
імені професора Володимира Бурячка

Освітньо-кваліфікаційних рівень – магістр
Спеціальність 125 Кібербезпека та захист інформації
Освітня програма 125.00.01 Безпека інформаційних і комунікаційних систем

«Затверджую»
Завідувач кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка
кандидат технічних наук, доцент
Складанний П.М.

_____
(підпис)
« ___ » _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Тімакову Андрію Івановичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження та вироблення рекомендацій щодо моделей автентифікації та авторизації в рамках методології Zero Trust;
керівник к.т.н., доц. Складанний Павло Миколайович
затверджені наказом ректора від «___»_____ 20___ року №__.
2. Термін подання студентом роботи «___»_____ 20___ р.
3. Вихідні дані до роботи:

3.1 науково-технічна та нормативна література з теми дослідження: науково-технічні праці - 52; нормативна література: NIST SP 800-207 — Zero Trust Architecture, ISO/IEC 27033-6:2016 — Network Security — Part 6: Securing inter-domain routing communications, ISO/IEC 27001:2022 — Information Security Management Systems — Requirements, Cloud Security Alliance (CSA) Zero Trust Guidelines;

3.2 методи: системний підхід, моделювання, методи теорії інформаційної безпеки, ризик-орієнтований аналіз, статистичні методи, методи нечіткої логіки, методи машинного навчання, експертне оцінювання, імітаційне моделювання;

3.3 технології: Zero Trust Architecture, Identity and Access Management (IAM), Multi-Factor Authentication (MFA), Single Sign-On (SSO), Security Information and Event Management (SIEM), Endpoint Detection and Response (EDR), TLS 1.3, OAuth 2.0;

3.4 алгоритми: RBAC, ABAC, PBAC, алгоритм оцінювання рівня довіри користувача, алгоритм автентифікації на основі багатофакторної перевірки (MFA), алгоритм перевірки токенів OAuth 2.0/OpenID Connect, алгоритм обчислення контекстного ризику доступу, алгоритм прийняття рішень у моделі Zero Trust;

3.5 мова програмування: Python (Flask, FastAPI, scikit-learn);

3.6 математичні моделі та методи: формалізована математична модель коефіцієнта довіри користувача, модель розрахунку довіри, адаптивний алгоритм автентифікації та авторизації, імітаційна модель політики довіри.

4. Зміст текстової частини роботи (перелік питань, які потрібно розробити):

4.1. Розглянути еволюцію моделей контролю доступу та теоретичні засади автентифікації й авторизації користувачів.

4.2. Провести класифікацію сучасних методів ідентифікації, автентифікації та багаторівневих моделей контролю доступу.

4.3. Дослідити концептуальні принципи методології Zero Trust та стандарти NIST SP 800-207, ISO/IEC 27033-6 і CSA Zero Trust Guidelines.

4.4. Розробити математичну модель розрахунку коефіцієнта довіри користувача на основі контекстних і поведінкових параметрів.

4.5. Побудувати схему потоків автентифікації та авторизації в архітектурі Zero Trust і провести імітаційне моделювання процесу прийняття політики доступу.

4.6. Провести тестування ефективності моделі за показниками продуктивності, точності та затримки при динамічній автентифікації.

4.7. Розробити практичні рекомендації щодо впровадження Zero Trust у корпоративних інформаційно-комунікаційних системах для підвищення рівня захищеності доступу та зниження ризиків внутрішніх і зовнішніх загроз.

5. Перелік графічного матеріалу:

5.1 Презентація доповіді, виконана в Microsoft PowerPoint.

5.2 Типові схеми: рисунків - 24.

6. Дата видачі завдання «___»_____ 20___ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів підготовки роботи	Термін виконання	Примітка
1.	Уточнення постановки завдання		
2.	Аналіз літератури		
3.	Обґрунтування вибору рішення		
4.	Збір даних		
5.	Виконання та оформлення розділу 1.		
6.	Виконання та оформлення розділу 2.		
7.	Виконання та оформлення розділу 3.		
8.	Вступ, висновки, реферат		
9.	Апробація роботи на науково-методичному семінарі та/або науково-технічній конференції		
10.	Оформлення та друк текстової частини роботи		
11.	Оформлення презентацій		
12.	Отримання рецензій		
13.	Попередній захист роботи		
14.	Захист в ЕК		

Студент _____ Тімаков Андрій Іванович
(підпис) (прізвище, ім’я, по батькові)

Науковий керівник _____ Складанний Павло Миколайович
(підпис) (прізвище, ім’я, по батькові)

РЕФЕРАТ

Кваліфікаційна робота присвячена дослідженню, аналізу та розробці рекомендацій щодо моделей автентифікації та авторизації в інформаційно-комунікаційних системах на основі методології Zero Trust.

Робота складається зі вступу, трьох розділів, що містять 28 рисунків та 5 таблиць, висновків та списку використаних джерел, що містить 56 найменувань. Загальний обсяг роботи становить 95 аркушів, а також додатки, перелік умовних скорочень.

Об'єктом дослідження є процес автентифікації та авторизації користувачів в інформаційно-комунікаційних системах у контексті забезпечення безпеки доступу на основі принципів Zero Trust.

Предметом дослідження є моделі, методи та механізми побудови систем автентифікації й авторизації в архітектурі Zero Trust, а також способи оцінювання довіри користувача та динамічного контролю доступу.

Метою роботи є дослідження та розроблення рекомендацій щодо вдосконалення моделей автентифікації та авторизації користувачів в інформаційно-комунікаційних системах на основі принципів методології Zero Trust для підвищення рівня безпеки доступу та мінімізації ризиків несанкціонованих дій.

Для досягнення поставленої мети у роботі: розглянуто еволюцію моделей контролю доступу та теоретичні засади автентифікації й авторизації користувачів; проведено класифікацію сучасних методів ідентифікації, автентифікації та багаторівневих моделей контролю доступу; досліджено концептуальні принципи методології Zero Trust та стандарти NIST SP 800-207, ISO/IEC 27033-6 і CSA Zero Trust Guidelines; розроблено математичну модель розрахунку коефіцієнта довіри користувача на основі контекстних і поведінкових параметрів; побудовано схему потоків автентифікації та авторизації в архітектурі Zero Trust і проведено імітаційне моделювання процесу прийняття політики доступу; проведено тестування ефективності моделі за показниками продуктивності, точності та затримки при динамічній автентифікації; розроблено практичні рекомендації щодо впровадження Zero Trust у корпоративних інформаційно-комунікаційних системах

для підвищення рівня захищеності доступу та зниження ризиків внутрішніх і зовнішніх загроз.

Наукова новизна одержаних результатів. Новими науково-обґрунтованими результатами, які отримані в роботі, є: розроблено формалізовану математичну модель коефіцієнта довіри користувача в архітектурі Zero Trust, яка враховує контекстні, поведінкові та технічні параметри доступу та забезпечує динамічне оновлення політик автентифікації; запропоновано адаптивний алгоритм автентифікації та авторизації з використанням принципів машинного навчання й нечіткої логіки, що дозволяє в режимі реального часу реагувати на зміну рівня ризику та підвищує стійкість системи до несанкціонованих дій; створено імітаційну модель політики довіри в середовищі Zero Trust, яка відображає взаємодію між користувачем, контекстом сесії та механізмами контролю доступу, забезпечуючи можливість оцінки ефективності системи за показниками точності, продуктивності та затримки. Додатково розроблено методику інтеграції моделей автентифікації та авторизації з архітектурними компонентами Policy Engine, Policy Administrator і Policy Enforcement Point, що забезпечує узгодженість політик безпеки на всіх рівнях системи. Запропоновано формальний підхід до оцінювання динамічної довіри, який базується на принципах Zero Trust і дозволяє кількісно вимірювати рівень ризику доступу для кожної сесії. Уперше сформовано узагальнену систему показників ефективності моделей автентифікації в рамках Zero Trust, що поєднує метричні, поведінкові та контекстні критерії для комплексної оцінки захищеності інформаційно-комунікаційних систем.

Галузь застосування. Результати роботи можуть бути використані для впровадження архітектури Zero Trust у корпоративних та державних інформаційно-комунікаційних системах з метою підвищення рівня захищеності доступу, динамічного управління автентифікацією і авторизацією користувачів та автоматизованого контролю подій безпеки відповідно до міжнародних стандартів. Крім того, отримані результати можуть бути інтегровані в системи управління ідентифікацією та доступом (IAM) і платформи моніторингу безпеки (SIEM/SOAR) для формування адаптивної політики доступу на основі ризику. Запропоновані

моделі забезпечують можливість контекстно-залежної перевірки довіри до суб’єктів і пристроїв, що сприяє зниженню ризику компрометації облікових даних. Практичне впровадження розроблених рекомендацій дозволяє створити масштабовану й стійку до загроз Zero Trust-інфраструктуру в межах сучасних цифрових підприємств.

Ключові слова: ZERO TRUST, АВТЕНТИФІКАЦІЯ, АВТОРИЗАЦІЯ, ДОВІРА КОРИСТУВАЧА, КОНТРОЛЬ ДОСТУПУ, КОНТЕКСТНА БЕЗПЕКА, MFA, SIEM, SOAR, ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНА СИСТЕМА.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ABAC – Attribute-Based Access Control — атрибутна модель контролю доступу

API – Application Programming Interface — програмний інтерфейс застосунку

DAC – Discretionary Access Control — дискреційна модель контролю доступу

AI – Artificial Intelligence — штучний інтелект

IAM – Identity and Access Management — система керування ідентифікацією та доступом

IdP – Identity Provider — постачальник ідентичностей

JWT – JSON Web Token — токен доступу у форматі JSON

MAC – Mandatory Access Control — мандатна модель контролю доступу

MFA – Multi-Factor Authentication — багатофакторна автентифікація

ML – Machine Learning — машинне навчання

OAuth 2.0 – протокол авторизації на основі токенів доступу

PBAC – Policy-Based Access Control — політико-орієнтована модель контролю доступу

PE – Policy Engine — рушій політик (модуль прийняття рішень)

PEP – Policy Enforcement Point — точка виконання політик

PDP – Policy Decision Point — точка прийняття рішень щодо доступу

SIEM – Security Information and Event Management — система моніторингу та кореляції подій безпеки

SOAR – Security Orchestration, Automation and Response — оркестрація, автоматизація та реагування на інциденти

SSO – Single Sign-On — єдиний вхід користувача

ZTNA – Zero Trust Network Access — мережевий доступ на основі Zero Trust

ZTA – Zero Trust Architecture — архітектура Zero Trust

ZT – Zero Trust — принцип «ніколи не довіряй, завжди перевіряй»

ВСТУП

Сучасні інформаційно-комунікаційні системи функціонують у середовищі підвищених кіберзагроз, де традиційні моделі контролю доступу — такі як DAC, MAC, RBAC або ABAC — виявляються недостатньо ефективними через статичність політик, відсутність контекстного аналізу та обмежену здатність реагувати на внутрішні загрози. Зростання кількості віддалених користувачів, використання хмарних і гібридних інфраструктур, а також інтенсифікація атак типу insider threat, phishing і session hijacking вимагають переходу до нової парадигми безпеки, що базується на постійному підтвердженні довіри. У цьому контексті актуальною науково-прикладною проблемою є розроблення моделей автентифікації та авторизації користувачів на основі принципів Zero Trust Architecture, які забезпечують динамічну оцінку ризиків, контекстну перевірку достовірності дій та автоматизоване оновлення політик доступу в режимі реального часу.

Актуальність роботи полягає в необхідності розробити сучасні моделі автентифікації та авторизації користувачів на основі принципів Zero Trust, які забезпечують динамічне управління доступом, контекстну оцінку довіри та підвищення рівня захищеності інформаційно-комунікаційних систем.

Метою роботи є дослідження та розроблення рекомендацій щодо вдосконалення моделей автентифікації та авторизації користувачів в інформаційно-комунікаційних системах на основі принципів методології Zero Trust для підвищення рівня безпеки доступу та мінімізації ризиків несанкціонованих дій.

Для досягнення поставленої мети були поставлені та вирішенні такі **завдання**:

1. Розглянуто еволюцію моделей контролю доступу та теоретичні засади автентифікації й авторизації користувачів;
2. Проведено класифікацію сучасних методів ідентифікації, автентифікації та багаторівневих моделей контролю доступу;
3. Досліджено концептуальні принципи методології Zero Trust та стандарти NIST SP 800-207, ISO/IEC 27033-6 і CSA Zero Trust Guidelines;

4. Розроблено математичну модель розрахунку коефіцієнта довіри користувача на основі контекстних і поведінкових параметрів;

5. Побудовано схему потоків автентифікації та авторизації в архітектурі zero trust і проведено імітаційне моделювання процесу прийняття політики доступу;

6. Проведено тестування ефективності моделі за показниками продуктивності, точності та затримки при динамічній автентифікації;

7. Розроблено практичні рекомендації щодо впровадження zero trust у корпоративних інформаційно-комунікаційних системах для підвищення рівня захищеності доступу та зниження ризиків внутрішніх і зовнішніх загроз.

Об’єктом дослідження є процес автентифікації та авторизації користувачів в інформаційно-комунікаційних системах у контексті забезпечення безпеки доступу на основі принципів Zero Trust.

Предметом дослідження є моделі, методи та механізми побудови систем автентифікації й авторизації в архітектурі Zero Trust, а також способи оцінювання довіри користувача та динамічного контролю доступу.

Методи дослідження. Для вирішення означених вище наукових завдань в роботі використано методи системного аналізу, моделювання процесів автентифікації та авторизації, теорії нечітких множин, машинного навчання, імітаційного моделювання, а також методи статистичного аналізу ефективності та надійності інформаційно-комунікаційних систем.

Наукова новизна одержаних результатів. Новими науково-обґрунтованими результатами, які отримані в роботі, є: розроблено формалізовану математичну модель коефіцієнта довіри користувача в архітектурі Zero Trust, яка враховує контекстні, поведінкові та технічні параметри доступу та забезпечує динамічне оновлення політик автентифікації; запропоновано адаптивний алгоритм автентифікації та авторизації з використанням принципів машинного навчання й нечіткої логіки, що дозволяє в режимі реального часу реагувати на зміну рівня ризику та підвищує стійкість системи до несанкціонованих дій; створено імітаційну модель політики довіри в середовищі Zero Trust, яка відображає взаємодію між користувачем, контекстом сесії та механізмами контролю доступу, забезпечуючи

можливість оцінки ефективності системи за показниками точності, продуктивності та затримки. Додатково розроблено методику інтеграції моделей автентифікації та авторизації з архітектурними компонентами Policy Engine, Policy Administrator і Policy Enforcement Point, що забезпечує узгодженість політик безпеки на всіх рівнях системи. Запропоновано формальний підхід до оцінювання динамічної довіри, який базується на принципах Zero Trust і дозволяє кількісно вимірювати рівень ризику доступу для кожної сесії. Уперше сформовано узагальнену систему показників ефективності моделей автентифікації в рамках Zero Trust, що поєднує метричні, поведінкові та контекстні критерії для комплексної оцінки захищеності інформаційно-комунікаційних систем.

Зв'язок роботи з науковими програмами, планами, темами. Напрям дослідження безпосередньо пов'язаний з реалізацією концепції Zero Trust Architecture, що передбачає побудову безпечного середовища доступу до інформаційно-комунікаційних систем шляхом постійної перевірки довіри, мінімізації прав користувачів і динамічного контролю політик автентифікації та авторизації відповідно до сучасних стандартів кібербезпеки. Кваліфікаційна робота виконана відповідно до планів наукової і науково-технічної діяльності Київського столичного університету імені Бориса Грінченка в рамках науково-дослідної роботи: «Методи та моделі забезпечення кібербезпеки інформаційних систем переробки інформації та функціональної безпеки програмно-технічних комплексів управління критичної інфраструктури» (№0122U200483, КУБГ, м. Київ).

Теоретичне та практичне значення. Нові наукові результати, отримані в роботі, у сукупності складають науково обґрунтовану основу для подальшого розвитку методів автентифікації та авторизації в архітектурі Zero Trust, що дозволяє формалізувати процес оцінювання довіри користувачів і підвищити ефективність управління доступом у динамічних інформаційно-комунікаційних середовищах. Практичне значення полягає у можливості використання розроблених моделей і рекомендацій для впровадження адаптивних механізмів контролю доступу в корпоративних і державних системах, інтеграції їх із засобами

моніторингу безпеки (SIEM/SOAR) та створення комплексних систем захисту інформації відповідно до вимог міжнародних стандартів.

Галузь застосування. Результати роботи можуть бути використані для впровадження архітектури Zero Trust у корпоративних та державних інформаційно-комунікаційних системах з метою підвищення рівня захищеності доступу, динамічного управління автентифікацією і авторизацією користувачів та автоматизованого контролю подій безпеки відповідно до міжнародних стандартів. Крім того, отримані результати можуть бути інтегровані в системи управління ідентифікацією та доступом (IAM) і платформи моніторингу безпеки (SIEM/SOAR) для формування адаптивної політики доступу на основі ризику. Запропоновані моделі забезпечують можливість контекстно-залежної перевірки довіри до суб’єктів і пристроїв, що сприяє зниженню ризику компрометації облікових даних. Практичне впровадження розроблених рекомендацій дозволяє створити масштабовану й стійку до загроз Zero Trust-інфраструктуру в межах сучасних цифрових підприємств.

Розділ 1. АНАЛІТИЧНИЙ ОГЛЯД МОДЕЛЕЙ АВТЕНТИФІКАЦІЇ ТА АВТОРИЗАЦІЇ

Аналітичний огляд моделей автентифікації та авторизації показує, що еволюція механізмів контролю доступу пройшла шлях від простих discretionary та mandatory моделей (DAC, MAC) до рольових (RBAC), атрибутних (ABAC) і політико-орієнтованих (PBAC) підходів, які дозволили гнучкіше визначати права користувачів у складних інформаційно-комунікаційних системах. Сучасні тенденції вказують на перехід до багаторівневих і контекстно-залежних методів автентифікації, що поєднують багатофакторні (MFA), безпарольні (passwordless) та поведінкові технології ідентифікації. Зростання кількості атак, спрямованих на компрометацію облікових даних, а також розвиток хмарних і гібридних інфраструктур зумовили потребу у нових моделях безпеки, де довіра не є сталою, а перевіряється на кожному етапі доступу. У цьому контексті формується методологія Zero Trust Architecture, що базується на принципі «ніколи не довіряй, завжди перевіряй» та орієнтується на мінімізацію прав доступу, постійну автентифікацію, моніторинг поведінки користувача і контекстне прийняття рішень. Застосування Zero Trust забезпечує підвищений рівень безпеки в корпоративних і державних системах, де автентифікація та авторизація набувають динамічного характеру, а управління доступом стає адаптивним і ризик-орієнтованим.

1.1. Еволюція моделей контролю доступу (DAC, MAC, RBAC, ABAC, PBAC)

Еволюція моделей контролю доступу в інформаційних системах відображає поступове вдосконалення підходів до управління правами користувачів, спрямоване на підвищення гнучкості, масштабованості та рівня безпеки [1-2, 4, 6, 8]. На ранніх етапах розвитку комп’ютерних систем основна увага приділялася простим механізмам обмеження доступу до ресурсів, які не враховували контекст взаємодії та рівень довіри до користувача чи середовища. З часом потреби

організацій у забезпеченні цілісності та конфіденційності інформації зумовили появу формалізованих моделей контролю доступу.

Першою була модель Discretionary Access Control (DAC) — дискреційний контроль доступу, у якій власник ресурсу самостійно визначає, хто може отримати доступ до даних. Вона забезпечувала простоту реалізації, однак мала суттєвий недолік: користувач із наданими правами міг передавати доступ іншим, що створювало ризики несанкціонованого поширення інформації [3, 6-7, 9-11]. Модель DAC була ефективною для невеликих систем, але виявилася вразливою у великих корпоративних мережах із багаторівневою структурою доступу.

На рис. 1.1 подано послідовну еволюцію моделей контролю доступу — від дискреційної (DAC) і мандатної (MAC) до рольової (RBAC), атрибутної (ABAC), політико-орієнтованої (PBAC) та сучасної парадигми Zero Trust. Кожна наступна модель відображає підвищення рівня гнучкості, контекстності та автоматизації прийняття рішень про доступ.

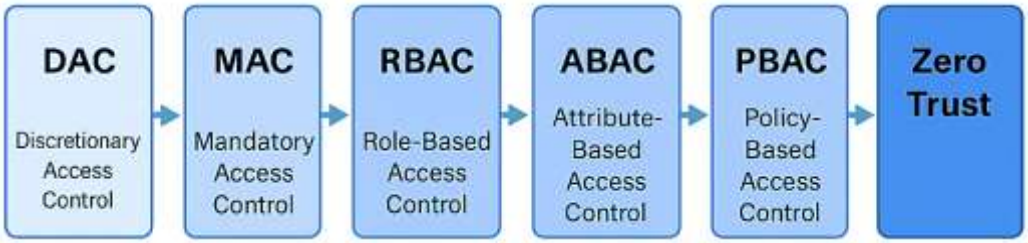

Рис. 1.1. Етапи еволюції моделей контролю доступу

Для підвищення рівня захищеності була запропонована Mandatory Access Control (MAC) — мандатна модель, у якій рішення про доступ приймає система, а не користувач. MAC передбачає наявність рівнів конфіденційності та правил, що визначають, хто і до яких ресурсів може мати доступ [3, 8, 12-14, 19-21]. Цей підхід забезпечив високу безпеку в середовищах з критично важливими даними, наприклад у військових або урядових структурах. Водночас жорсткість правил і складність адміністрування робили модель малопридатною для гнучких корпоративних систем.

Подальший розвиток отримала Role-Based Access Control (RBAC) — модель, у якій доступ визначається не на рівні користувача, а на рівні ролей, що

відповідають посадовим або функціональним обов'язкам. Кожна роль має певний набір прав, а користувач отримує їх автоматично при призначенні ролі. RBAC значно спростила адміністрування доступу та підвищила керованість безпеки в організаціях із великою кількістю співробітників. Однак вона залишалася статичною — зміни контексту або поведінки користувача не враховувалися в реальному часі.

На рис. 1.2 представлено структурну схему моделі Policy-Based Access Control (PBAC), яка демонструє процес прийняття рішень щодо доступу на основі політик безпеки. Користувач надсилає запит на доступ до ресурсу, після чого запит опрацьовується через PEP (Точку виконання політики), PDP (Точку прийняття рішення) та PAP (Точку адміністрування політик). У процесі прийняття рішення враховуються дані з джерел контекстної інформації — час, місце, стан пристрою, рівень ризику тощо [10-11, 15-18, 22, 25]. Схема відображає взаємозв'язок між компонентами моделі та підкреслює динамічний характер управління доступом, що є основою для реалізації принципів архітектури Zero Trust.

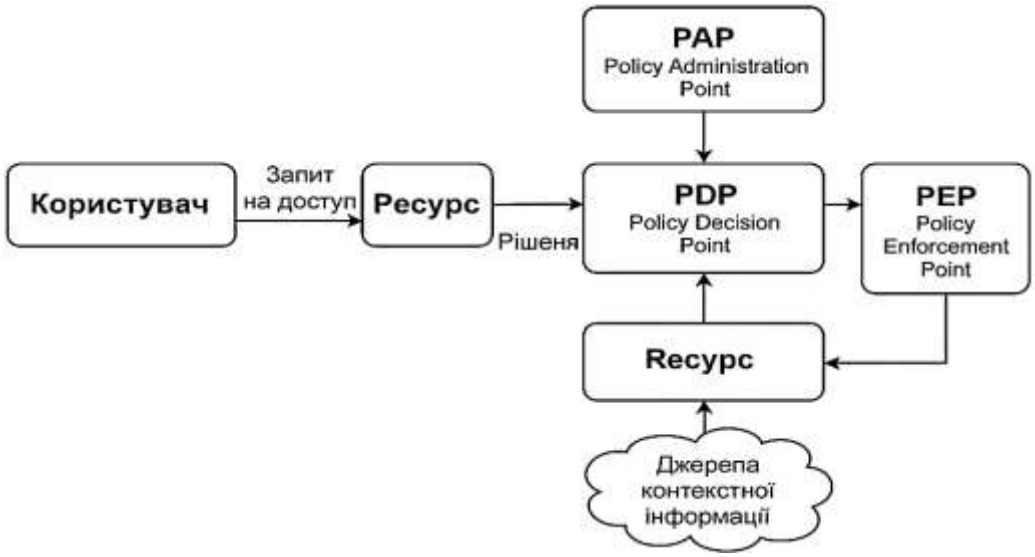

Рис. 1.2. Структурна схема моделі Policy-Based Access Control (PBAC)

Для вирішення цієї проблеми була розроблена Attribute-Based Access Control (ABAC) — атрибутна модель, яка приймає рішення про доступ на основі сукупності атрибутів користувача (посада, рівень доступу, пристрій), ресурсу (тип даних, конфіденційність) і середовища (час, місце, стан мережі) [1, 3-4, 23-24].

Такий підхід забезпечив високу гнучкість і динамічність, дозволивши враховувати поточний контекст. Однак із ростом складності політик і збільшенням кількості атрибутів адміністрування стало ресурсомістким, а моделювання поведінки користувачів — складним завданням.

Еволюційним розвитком ABAC стала Policy-Based Access Control (PBAC) — політико-орієнтована модель, у якій рішення про доступ приймається на основі набору формалізованих політик, що можуть враховувати контекст, рівень ризику, історію активності користувача та поточний стан системи. PBAC поєднує принципи атрибутного підходу й динамічного управління політиками, що дозволяє реалізувати концепцію безперервного контролю доступу [2, 6, 8, 17, 27]. Саме на основі PBAC сформувалася методологія Zero Trust Architecture (ZTA), яка відмовляється від традиційного поняття «довіреної зони» та передбачає постійну перевірку автентичності й авторизації кожної дії користувача або пристрою.

Таблиця 1.1

Порівняльна характеристика основних моделей контролю доступу

Критерій	DAC	MAC	RBAC	ABAC	PBAC
Централізація управління	Відсутня, рішення приймає власник ресурсу	Повна централізація, рішення приймає система	Централізоване керування ролями	Централізоване, але залежить від атрибутів	Централізоване, керується політиками безпеки
Гнучкість	Висока, але без контролю	Низька, жорстка ієрархія	Середня, залежить від ролей	Висока, завдяки атрибутам	Дуже висока, політики адаптуються до контексту
Контекстність	Відсутня	Мінімальна	Обмежена (ролі фіксовані)	Висока (атрибути середовища, часу тощо)	Дуже висока, враховує ризики, довіру й поведінку
Масштабованість	Обмежена, складне адміністрування	Низька	Висока у великих системах	Висока, але складна конфігурація	Висока, політики автоматизовані
Рівень безпеки	Низький, ризик розповсюдження прав	Високий, контроль системою	Високий при належному адмініструванні	Високий, контекстна перевірка	Дуже високий, динамічна перевірка довіри (Zero Trust)

На рис. 1.3 показано логічний зв’язок між моделлю PBAC та архітектурою Zero Trust, у якій компоненти Policy Engine, Policy Administrator і Policy

Enforcement Point утворюють єдиний контур управління доступом. Policy Engine аналізує політики й контекст довіри, Policy Administrator реалізує рішення та оновлює правила доступу, а Policy Enforcement Point виконує перевірку запитів користувачів. Схема демонструє принцип «ніколи не довіряй, завжди перевіряй» та підкреслює роль PBAC як ядра Zero Trust Architecture.

Рис. 1.3. Взаємозв’язок між PBAC і архітектурою Zero Trust

Таким чином, розвиток моделей контролю доступу демонструє чітку тенденцію до переходу від статичних, централізованих рішень до гнучких, контекстно-орієнтованих систем, здатних адаптуватися до динамічних умов середовища. Модель PBAC стала теоретичним та практичним фундаментом для архітектури Zero Trust, у межах якої контроль доступу перетворюється на безперервний процес оцінки довіри, а прийняття рішень відбувається з урахуванням поточного рівня ризику та поведінкових характеристик користувача.

1.2. Сучасні методи ідентифікації та автентифікації (MFA, SSO, passwordless, biometrics)

Розвиток цифрових технологій і масова міграція бізнес-процесів у хмарні середовища зумовили потребу у створенні більш гнучких і надійних механізмів ідентифікації та автентифікації користувачів [2, 6-8, 10]. Традиційна модель «логін–пароль» втрачає ефективність через низьку стійкість до фішингових атак, повторне використання паролів і людський фактор, який залишається головною причиною компрометації облікових даних. Це стимулювало перехід до багатофакторних і безпарольних систем автентифікації, здатних поєднувати зручність і високий рівень безпеки.

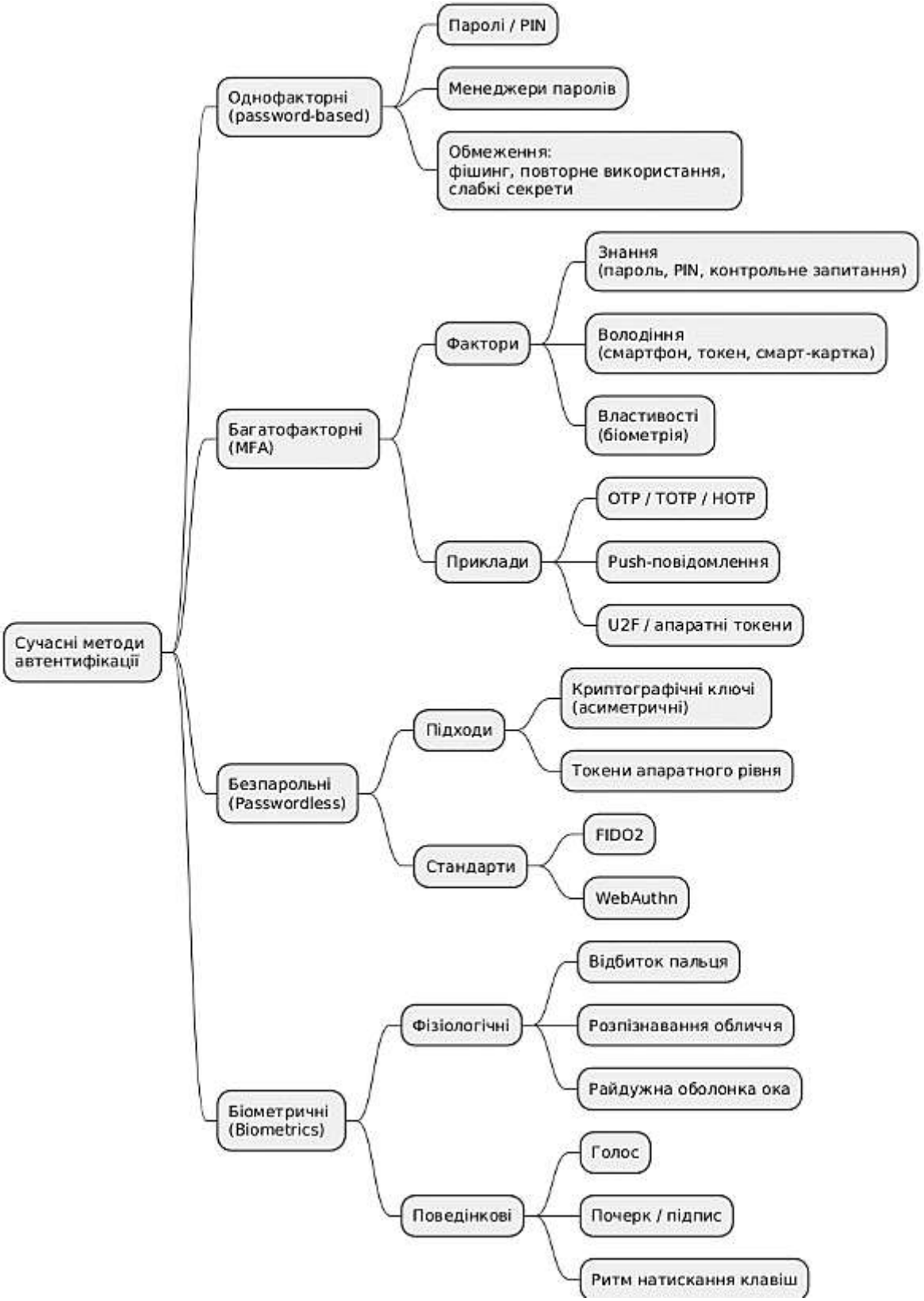

Рис. 1.4. Ієрархічна класифікація сучасних методів автентифікації користувачів

На рис. 1.4 представлено ієрархічну класифікацію сучасних методів автентифікації користувачів, побудовану у вигляді ментальної карти. Вона поділяє

всі підходи до підтвердження особи на чотири основні групи: однофакторні, багатофакторні (MFA), безпарольні (Passwordless) та біометричні (Biometrics). У межах кожної групи відображено ключові приклади та технології. Однофакторні методи базуються на знанні користувача (паролі, PIN-коди), проте мають обмеження щодо стійкості до фішингу [17, 25-27, 30-31]. Багатофакторна автентифікація поєднує кілька незалежних факторів — знання, володіння й властивості, забезпечуючи вищий рівень захисту. Безпарольні системи ґрунтуються на криптографічних ключах і токенах апаратного рівня, що реалізують стандарти FIDO2 та WebAuthn [15, 38-39]. Біометричні методи поділяються на фізіологічні (відбиток пальця, обличчя, райдужна оболонка ока) та поведінкові (голос, почерк, ритм натискання клавіш), забезпечуючи індивідуалізовану перевірку користувача.

Діаграма узагальнює еволюцію підходів до автентифікації, демонструючи перехід від традиційних паролів до контекстно-залежних, безпарольних і біометричних рішень, що є фундаментом для реалізації концепції Zero Trust Architecture.

Одним із найпоширеніших сучасних підходів є Multi-Factor Authentication (MFA) — багатофакторна автентифікація, що базується на використанні двох або більше незалежних факторів: знання (пароль або PIN), володіння (смартфон, токен, смарт-карта) та властивості (біометричні дані користувача). Використання MFA істотно знижує ризик несанкціонованого доступу, навіть якщо один із факторів було скомпрометовано [8, 25, 27, 40]. У сучасних корпоративних середовищах все частіше впроваджуються адаптивні MFA-рішення, які враховують контекст доступу — місцезнаходження користувача, тип пристрою або час входу до системи.

Ще одним популярним підходом є Single Sign-On (SSO) — технологія єдиного входу, що дозволяє користувачу автентифікуватися один раз для отримання доступу до кількох взаємопов’язаних сервісів [14, 45, 50-51]. Це підвищує зручність роботи й зменшує кількість облікових даних, які потрібно запам’ятовувати, однак створює додаткові вимоги до захисту центрального вузла автентифікації. Сучасні

системи SSO інтегруються з протоколами OAuth 2.0, OpenID Connect та SAML, забезпечуючи сумісність із хмарними платформами та мобільними додатками. У контексті архітектури Zero Trust SSO поєднується з багатофакторною автентифікацією та механізмами перевірки контексту доступу для мінімізації ризиків компрометації токенів [19-20]. Такий підхід дозволяє реалізувати концепцію «мінімально необхідних привілеїв» (Least Privilege) і забезпечує централізований контроль доступу без зниження зручності користувачів.

Інноваційним напрямом розвитку автентифікації є passwordless-підхід, який повністю усуває використання паролів як основного засобу автентифікації. Такі системи ґрунтуються на криптографічних ключах, біометрії або токенах апаратного рівня (наприклад, FIDO2, WebAuthn), що дозволяє досягти високої стійкості до фішингу та атакам типу *man-in-the-middle* [43, 46]. Безпарольна автентифікація є одним із базових елементів архітектури Zero Trust, адже вона зменшує ризики компрометації ідентифікаційних даних.

Її впровадження сприяє переходу до моделі безперервної перевірки довіри (Continuous Authentication), де автентифікація здійснюється не одноразово, а протягом усієї сесії користувача. Поєднання passwordless-технологій із поведінковою біометрією та оцінкою контекстних атрибутів (місце, пристрій, час, рівень ризику) забезпечує адаптивність і персоналізацію доступу. У межах архітектури Zero Trust такі механізми інтегруються з політиками динамічного прийняття рішень Policy Engine, що підвищує точність контролю та знижує кількість хибних спрацьовувань [22, 24]. Перспективним є поєднання безпарольної автентифікації з технологіями багатофакторної довіри та розподіленими ідентифікаторами (Decentralized Identifiers, DID) для створення повністю безпечної та прозорої моделі доступу.

На рис. 1.5 показано послідовність етапів багатофакторної автентифікації, що реалізується в архітектурі Zero Trust. Користувач вводить логін і пароль, після чого служба автентифікації перевіряє контекст — зокрема час, пристрій та геолокацію. Якщо політика безпеки вимагає додаткового підтвердження, ініціюється другий фактор (SMS-код, push-повідомлення, токен або біометрія). Після успішної

перевірки обох факторів Policy Engine приймає остаточне рішення про надання доступу до ресурсу. Схема ілюструє контекстно-залежну логіку автентифікації, що є ключовим принципом Zero Trust.

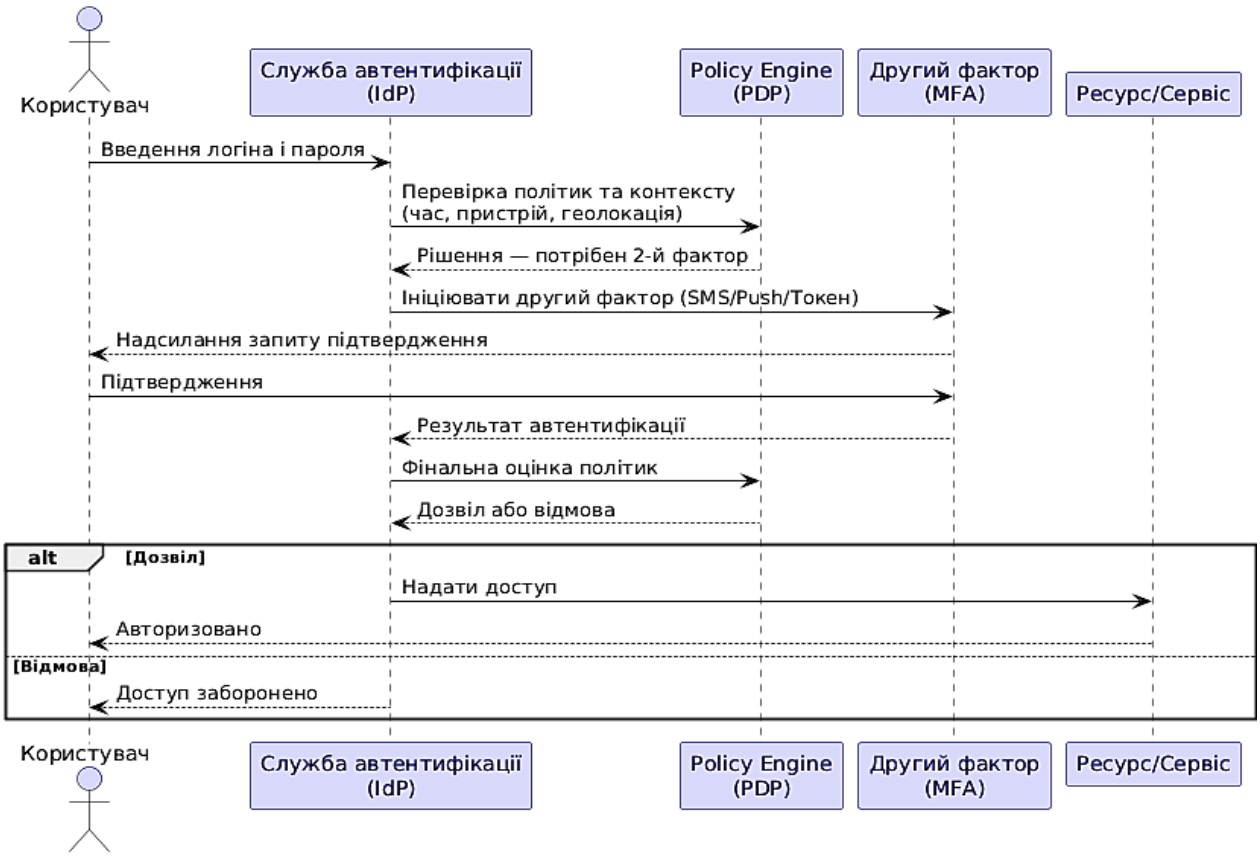

Рис. 1.5. Схема процесу багатофакторної автентифікації користувача

Окрему роль відіграють біометричні методи автентифікації, які використовують унікальні фізіологічні або поведінкові характеристики користувача: відбитки пальців, розпізнавання обличчя, райдужної оболонки ока, голосу чи патернів натискання клавіш. Біометрія дозволяє забезпечити високий рівень зручності та ідентифікаційної точності, проте вимагає ретельного дотримання вимог конфіденційності та захисту персональних даних.

Сучасні системи автентифікації дедалі частіше комбінують зазначені підходи, створюючи адаптивні моделі доступу, які враховують контекст, ризики та поведінку користувача. Такі рішення лягають в основу концепції Zero Trust, де автентифікація перестає бути одноразовою процедурою і перетворюється на безперервний процес підтвердження довіри.

1.3. Технології контекстної та поведінкової автентифікації (risk-based, continuous, adaptive)

Сучасні підходи до забезпечення автентифікації поступово переходять від статичних моделей до контекстних та поведінкових технологій, що ґрунтуються на оцінці ризику та динамічному аналізі дій користувача.

Контекстна автентифікація (risk-based authentication) передбачає врахування додаткових параметрів середовища під час спроби входу: геолокації, часу доби, типу пристрою, мережевих характеристик та історії попередніх сесій. Якщо система виявляє відхилення від звичного профілю користувача (наприклад, спробу входу з іншої країни або з нового пристрою), рівень ризику зростає, і користувачеві пропонується пройти додаткову перевірку [19, 20]. Таким чином, рішення про доступ приймається не лише за фактом введення правильних облікових даних, а й з урахуванням контексту їх використання.

Поведінкова автентифікація (behavioral or continuous authentication) орієнтується на аналіз індивідуальних моделей поведінки користувача під час роботи з системою. До таких ознак належать швидкість набору тексту, ритм натискання клавіш, характер рухів миші, темп прокрутки сторінок, а також біометричні патерни (положення пристрою, хода тощо) [10-11]. Ці дані обробляються алгоритмами машинного навчання, які створюють унікальний поведінковий профіль користувача [21, 24]. Якщо виявляються аномалії, система ініціює повторну перевірку або тимчасово обмежує доступ.

Адаптивна автентифікація (adaptive authentication) є інтегрованим підходом, що поєднує контекстні та поведінкові фактори в єдину динамічну модель оцінки довіри. Вона дозволяє системі автоматично змінювати рівень перевірки залежно від ризику: наприклад, у безпечному середовищі може застосовуватись лише один фактор, тоді як за підвищеного ризику — кілька послідовних етапів підтвердження. Такі рішення використовують алгоритми машинного навчання для аналізу поведінкових шаблонів користувачів і виявлення аномалій у процесі входу чи взаємодії з ресурсами [38, 50]. Завдяки цьому система здатна в режимі реального

часу реагувати на зміни ризикового профілю та коригувати політику доступу без втручання адміністратора [51]. Адаптивна автентифікація є ключовим компонентом Zero Trust, оскільки забезпечує баланс між зручністю користувача та високим рівнем захищеності інформаційно-комунікаційної системи.

На рис. 1.6 представлено вертикальну DFD-модель, що відображає взаємодію між користувачем, системою автентифікації та модулями Zero Trust. Послідовно показано обмін даними від моменту введення облікових даних до ухвалення рішення про доступ. Система аналізує контекстні параметри (час, пристрій, геолокацію) та поведінкові характеристики користувача, передаючи їх до Policy Engine, який оцінює рівень ризику й формує адаптивні вимоги, наприклад застосування додаткових факторів перевірки. Результати аналізу визначають, чи буде користувачу надано доступ до ресурсу. Діаграма демонструє, як поєднання контекстної, поведінкової та адаптивної автентифікації забезпечує безперервну оцінку довіри в середовищі Zero Trust Architecture.

1.6. Модель інтеграції контекстної, поведінкової та адаптивної автентифікації в архітектурі Zero Trust

Такі технології є фундаментом Zero Trust Architecture, де жоден користувач чи пристрій не вважається надійним за замовчуванням, а доступ надається лише після динамічної оцінки контексту та поведінки. Це дозволяє мінімізувати ризики

компрометації облікових даних і підвищити загальну стійкість системи до сучасних кіберзагроз.

1.4. Недоліки класичних моделей доступу в умовах гібридних інфраструктур

Класичні моделі контролю доступу — DAC, MAC, RBAC і навіть гнучкіша ABAC — створювалися для статичних середовищ із єдиною довіреною мережею. Перехід до гібридних інфраструктур із хмарними сервісами та мобільними платформами виявив їхні ключові обмеження [10-12, 50]. Основна проблема — статичність правил доступу: у DAC і RBAC права визначаються наперед і не враховують змін контексту, що ускладнює дотримання принципу *Least Privilege*. Також відсутня контекстна оцінка ризику — моделі не враховують поведінкові аномалії чи стан пристрою, що створює ризики компрометації.

Крім того, вони погано масштабуються в розподілених хмарних середовищах через дублювання політик, несумісність систем і складність аудиту. Відсутність єдиного механізму перевірки автентичності пристрою призводить до підвищеного ризику атак типу device spoofing або session hijacking. Таким чином, класичні підходи до контролю доступу не здатні забезпечити необхідний рівень безпеки в умовах гібридної, динамічної інфраструктури.

Отже, традиційні підходи не відповідають вимогам динамічних інфраструктур, що зумовлює перехід до Zero Trust Architecture, де доступ визначається за багатофакторною, контекстно-залежною оцінкою довіри в реальному часі [2, 4, 6]. Таким чином, класичні підходи до контролю доступу не здатні забезпечити необхідний рівень безпеки в умовах гібридної, динамічної інфраструктури. Це обґрунтовує необхідність переходу до Zero Trust Architecture, у якій доступ базується не лише на ідентичності користувача, а й на багатофакторній, контекстно-залежній оцінці довіри, що оновлюється в режимі реального часу.

1.5. Передумови становлення та сучасні підходи до концепції Zero Trust Architecture

Концепція Zero Trust Architecture (ZTA) виникла як реакція на обмеження традиційних моделей безпеки, що ґрунтувалися на периметральному захисті. Раніше корпоративні системи діяли в межах чітко визначеного периметра, де внутрішнім користувачам і пристроям довіряли. Однак розвиток хмарних сервісів, мобільних технологій, гібридних інфраструктур і моделей віддаленої роботи призвів до розмиття цього периметра та зробив підхід «довіряй, але перевіряй» неефективним [6, 11]. Внаслідок цього класичні моделі доступу (DAC, MAC, RBAC, ABAC) виявилися нездатними гарантувати безпеку в динамічному цифровому середовищі, оскільки не враховують контекст запиту, поведінку користувача та рівень ризику взаємодії (табл. 1.2).

Таблиця 1.2

Порівняння традиційної моделі безпеки та концепції Zero Trust

Критерій	Традиційна модель безпеки	Zero Trust Architecture
Довіра	Ґрунтується на периметральному принципі «довіряй внутрішньому»	Довіра динамічна, постійно перевіряється на кожному етапі
Контроль доступу	Одноразова автентифікація при вході	Безперервна перевірка контексту, поведінки та ризику
Масштабованість	Обмежена, складно інтегрувати нові сервіси й пристрої	Гнучка, орієнтована на хмарні та гібридні середовища
Перевірка користувача	Залежить від статичних правил і ролей	Використовує адаптивні, поведінкові та контекстні фактори
Захист даних	Зосереджений на мережевому периметрі	Реалізований на рівні кожного користувача, пристрою та ресурсу

Табл. 1.2 ілюструє відмінності між традиційною моделлю безпеки та архітектурою Zero Trust за ключовими критеріями. Вона показує перехід від статичної, периметральної довіри до динамічної, контекстно-залежної перевірки користувачів і пристроїв у реальному часі. Порівняльний аналіз демонструє, що у Zero Trust кожен запит доступу розглядається як потенційно недовірений, навіть якщо він надходить із внутрішньої мережі. Це забезпечує гнучке, багаторівневе управління безпекою, засноване на принципах мінімальних привілеїв, постійної автентифікації та моніторингу поведінки суб’єктів доступу.

Потреба у новій парадигмі безпеки зумовила появу принципу Zero Trust, запропонованого аналітиком Forrester Research Джоном Кіндервагом у 2010 році. Його суть виражається у постулаті *«Never trust, always verify»* — ніколи не довіряй, завжди перевіряй. У межах цієї концепції жоден користувач чи пристрій не вважається надійним за замовчуванням, а кожен запит доступу супроводжується перевіркою ідентичності, контексту та ризику [3-4, 14]. Довіра тут розглядається як динамічна змінна, що оновлюється залежно від поведінки користувача та умов середовища.

Рис. 1.7 ілюструє просторове порівняння основних моделей контролю доступу за чотирма критеріями — гнучкість, масштабованість, контекстність та безпека. Як видно з діаграми, моделі PBAC і особливо Zero Trust демонструють найвищі інтегральні показники, забезпечуючи одночасно високий рівень безпеки та адаптивність до контексту. У той час як DAC і MAC мають обмежену гнучкість, сучасні підходи, засновані на політиках і контексті, формують більш збалансовану та ефективну систему управління доступом.

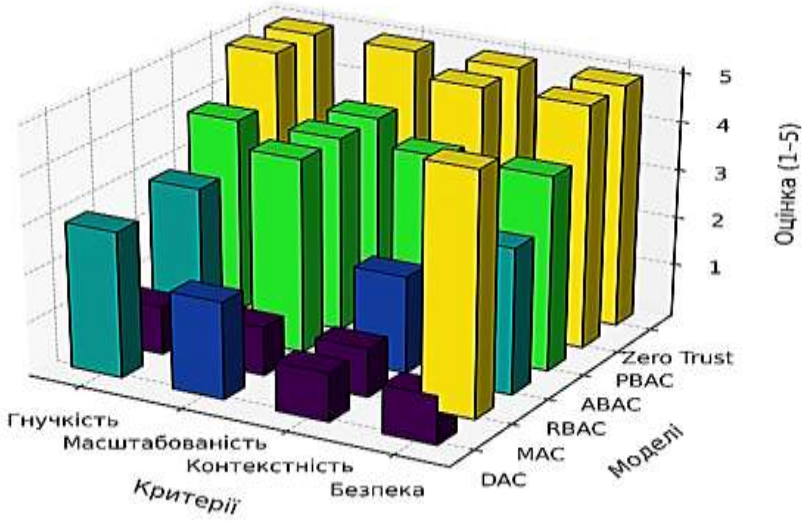

Рис. 1.7. Порівняння моделей контролю доступу

Діаграма на рис. 1.8 відображає логіку обміну повідомленнями між основними елементами Zero Trust — користувачем, точкою контролю політик (PEP), точкою прийняття рішень (PDP), адміністратором політик (PA) і цільовим ресурсом. Користувач ініціює запит доступу, який передається до PEP для

перевірки. PEP надсилає запит авторизації до PDP, що звертається до PA з метою отримання політики доступу. Після обробки запиту PDP повертає рішення (дозвіл або відмову), яке PEP застосовує до ресурсу, забезпечуючи контроль доступу відповідно до політик Zero Trust. Така схема ілюструє безперервний процес оцінки довіри, характерний для концепції Zero Trust Architecture. Цей процес гарантує, що кожен запит перевіряється окремо, незалежно від попередніх сеансів чи рівня довіри користувача. Завдяки такому підходу мінімізується ризик несанкціонованого доступу та підвищується стійкість системи до внутрішніх і зовнішніх загроз.

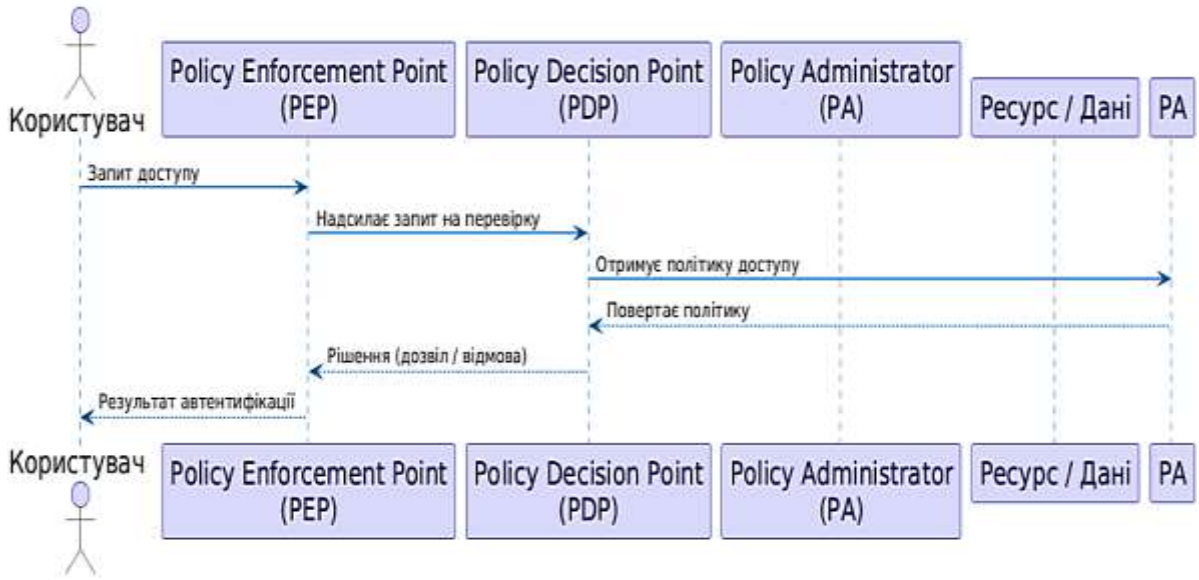

Рис. 1.8. Послідовність взаємодії компонентів архітектури Zero Trust під час перевірки доступу

Основні принципи Zero Trust включають мінімізацію привілеїв (Least Privilege Access), безперервну перевірку довіри (Continuous Verification), контекстно-залежне прийняття рішень (Context-Aware Access) та мікросегментацію мережі (Microsegmentation). Усі ці механізми реалізуються через ключові компоненти архітектури — Policy Decision Point (PDP), який оцінює рівень довіри, Policy Enforcement Point (PEP), що забезпечує виконання рішень на рівні доступу, та Policy Administrator (PA), який керує політиками ідентифікації, автентифікації й авторизації. Відповідно до стандарту NIST SP 800-207, Zero Trust Architecture поєднує технічні та організаційні аспекти: моніторинг трафіку, аналітику

поведінки, динамічне управління політиками, контроль відповідності пристроїв і аудит дій користувачів [12-15, 19-20, 51]. Аналогічні підходи визначено у ISO/IEC 27033-6:2016 та рекомендаціях Cloud Security Alliance (CSA) Zero Trust Guidelines [55-56], які описують практичні методи адаптації ZTA для хмарних і гібридних середовищ.

У сучасних наукових дослідженнях Zero Trust розглядається не лише як набір принципів, а як інтелектуальна модель динамічного управління довірою, що інтегрує штучний інтелект і машинне навчання [10-11, 16, 21-22]. Так, у працях Lee & Yang (2021) та Zhang & Liu (2022) розроблено алгоритми прогнозування ризику з урахуванням поведінкових і контекстних факторів користувача, а Wang & Chen (2021) запропонували застосування нечітких когнітивних карт (Fuzzy Cognitive Maps) для адаптивної оцінки довіри в реальному часі [18, 31]. Tan & Zhou (2021) показали ефективність нейромережевих моделей у виявленні аномальної активності під час доступу до корпоративних ресурсів [21, 48]. У практичній площині провідні компанії впроваджують Zero Trust у свої рішення: Google BeyondCorp відмовляється від VPN, застосовуючи контекстну автентифікацію; Microsoft Entra ID (Azure AD) реалізує адаптивний контроль на основі ризику; Palo Alto Prisma Access і Zscaler Zero Trust Exchange забезпечують хмарну мікросегментацію й захист користувачів незалежно від місця входу в систему.

Отже, концепція Zero Trust Architecture стала наступним етапом еволюції корпоративної безпеки, що поєднує аналітику контексту, поведінкове моделювання та адаптивну авторизацію в єдину систему безперервної перевірки довіри. Вона створює архітектурну основу для побудови самонавчальних і гнучких систем захисту, які здатні адаптувати політики доступу до поточних умов, мінімізувати ризики компрометації та забезпечувати цілісність інформаційного середовища підприємства.

Висновки до першого розділу

Проведений аналітичний огляд доводить, що сучасна еволюція систем автентифікації та авторизації є результатом поступового переходу від статичних і централізованих механізмів контролю доступу до динамічних, контекстно- та ризик-орієнтованих моделей. Традиційні підходи — DAC, MAC, RBAC і ABAC — створювали фундамент для побудови формалізованих систем управління доступом, проте вони не враховували контекст користувача, поведінкові фактори та зміни середовища в реальному часі [54]. Це зумовило появу більш гнучкої політико-орієнтованої моделі PBAC, яка інтегрує атрибути, політики та контекстні сигнали для прийняття рішень про доступ. Саме PBAC стала теоретичною основою концепції Zero Trust Architecture (ZTA), де довіра розглядається не як сталий атрибут, а як динамічна змінна, що постійно переоцінюється залежно від поведінки, місця, часу та ризиків.

Розвиток цифрової трансформації, хмарних технологій і мобільних середовищ суттєво змінив вимоги до автентифікації користувачів. У цьому контексті зросла роль багатофакторної (MFA), безпарольної (passwordless) та біометричної автентифікації, які забезпечують вищий рівень захисту облікових даних. Додатково, контекстна, поведінкова та адаптивна автентифікація забезпечують постійну оцінку довіри, враховуючи ризики, тип пристрою, геолокацію, а також моделі поведінки користувача. Такі технології формують ядро інтелектуальних систем доступу, здатних самостійно коригувати політики залежно від поточного рівня загроз.

Водночас аналіз виявив низку обмежень класичних моделей доступу, особливо в умовах гібридних інфраструктур, де зростає кількість взаємодій між локальними й хмарними середовищами. Проблеми масштабування, дублювання політик, відсутність єдиних механізмів перевірки автентичності пристроїв і неузгодженість форматів журналів безпеки ускладнюють забезпечення цілісності системи. Це підтверджує необхідність переходу від периметрального до Zero Trust-підходу, що базується на принципах «ніколи не довіряй, завжди перевіряй» і «найменших привілеїв» (Least Privilege Access).

Таким чином, розділ 1 підсумовує, що сучасна парадигма інформаційної безпеки переходить від статичних моделей до динамічних систем довіри, де автентифікація та авторизація інтегруються в безперервний процес оцінки ризику. Це створює основу для подальшого розвитку архітектури Zero Trust, у якій забезпечується адаптивний, інтелектуальний і самонавчальний контроль доступу, орієнтований на реальний рівень загроз і поведінку користувачів.

Розділ 2. ТЕОРЕТИЧНІ ОСНОВИ ТА МЕТОДОЛОГІЯ ДОСЛІДЖЕННЯ ZERO TRUST ARCHITECTURE

Концепція Zero Trust Architecture (ZTA) ґрунтується на відмові від традиційного периметрального захисту та базується на принципах «ніколи не довіряй, завжди перевіряй» і «мінімальних привілеїв». Вона передбачає, що кожен користувач, пристрій чи сервіс може бути потенційно скомпрометованим, тому доступ дозволяється лише після підтвердження автентичності та оцінки контексту в реальному часі. У стандартах NIST SP 800-207 Zero Trust визначається як динамічна модель безпеки, де рішення про доступ приймаються з урахуванням ризику, поведінки користувача, стану пристрою та поточної загрози.

Теоретичну основу ZTA становлять моделі PBAC (Policy-Based Access Control) та risk-based authentication, які поєднують контекст, атрибути, політики й оцінку довіри в єдиній системі управління доступом. Важливим елементом архітектури є компоненти Policy Engine, Policy Administrator та Policy Enforcement Point, що забезпечують ухвалення, адміністрування й виконання політик доступу відповідно до заданих критеріїв довіри.

Методологія дослідження Zero Trust базується на системному, ризик-орієнтованому та аналітичному підходах. Використано методи структурного аналізу, моделювання ризиків за ISO/IEC 27005 і NIST SP 800-30, когнітивного моделювання на основі нечіткої логіки та машинного навчання для виявлення аномалій і прогнозування інцидентів [55-56]. Такий підхід дозволяє оцінювати рівень довіри динамічно, з урахуванням поведінки користувача, типу пристрою та умов середовища.

Результатом дослідження є побудова багаторівневої моделі Zero Trust, що включає збір контексту, оцінку довіри та ризику і управління політиками доступу. Запропонована методологія забезпечує безперервну перевірку автентичності, адаптивне управління привілеями й підвищення стійкості системи до сучасних кіберзагроз.

2.1. Принципи Zero Trust: «Never trust, always verify»

Принцип «Never trust, always verify» є фундаментом концепції Zero Trust Architecture і означає повну відмову від автоматичної довіри до будь-якого користувача, пристрою чи сервісу, навіть якщо вони перебувають усередині корпоративної мережі. Кожен запит на доступ до ресурсу розглядається як потенційно небезпечний і проходить повну перевірку автентичності, авторизації та контексту в режимі реального часу [2, 6, 11]. Цей принцип спрямований на мінімізацію ризиків, пов’язаних із компрометацією облікових даних, зловмисними інсайдерами чи використанням вразливих пристроїв. У межах Zero Trust автентифікація й авторизація не є одноразовими подіями, а виконуються безперервно під час усієї сесії взаємодії користувача з системою. Контроль доступу базується на багатьох факторах — рівні ризику, поведінці користувача, стані пристрою та відповідності політикам безпеки.

У межах цього підходу довіра не є постійною — вона формується динамічно, щоразу, коли користувач взаємодіє з системою. Перевіряються не лише логін і пароль, а й поведінкові характеристики користувача, стан пристрою, геолокація, тип мережевого з’єднання та рівень поточного ризику. Рішення про доступ приймається на основі аналізу багатьох факторів, що дозволяє виявляти підозрілу активність ще до того, як вона призведе до інциденту.

Динамічна модель довіри передбачає, що кожна взаємодія користувача з системою супроводжується новою оцінкою ризику, а попередні рішення не мають довготривалої сили. Якщо виявляється зміна контексту — новий пристрій, незвична активність або підозріла мережа — рівень довіри автоматично знижується [8, 28-29]. Система може активувати додаткові механізми підтвердження, наприклад багатофакторну перевірку чи тимчасове обмеження доступу. Усе це відбувається у реальному часі, що дозволяє запобігти несанкціонованим діям ще до виникнення інциденту.

Практична реалізація принципу «ніколи не довіряй, завжди перевіряй» забезпечується за допомогою безперервної автентифікації (continuous

authentication), контекстно-залежного контролю доступу (context-aware access control) та оцінки ризику в реальному часі (real-time risk assessment). Наприклад, якщо користувач входить у систему з нового пристрою або з незвичної геолокації, система може вимагати додатковий фактор підтвердження (MFA) чи обмежити права доступу.

Крім того, сучасні системи Zero Trust використовують телеметрію пристроїв, аналізуючи рівень їх відповідності політикам безпеки, наявність оновлень, антивірусного захисту та сертифікатів. Дані з різних джерел — журналів подій, SIEM-платформ, систем моніторингу трафіку — агрегуються для побудови динамічного профілю довіри користувача. На основі цього профілю Policy Engine приймає рішення про надання, обмеження або тимчасове блокування доступу [6, 11]. У разі виявлення підозрілої поведінки, наприклад різкого зростання кількості запитів або зміни моделей активності, система автоматично знижує рівень довіри та активує додаткові перевірки. Таким чином, автентифікація перетворюється з одноразової процедури на безперервний процес оцінки ризику, що адаптується до контексту взаємодії. Такий механізм забезпечує не лише підвищений рівень безпеки, а й дозволяє зберегти зручність користувача, оскільки перевірки виконуються непомітно та автоматично, без необхідності постійного введення даних (табл. 2.1).

Таблиця 2.1

Джерела даних і їх вплив на рішення про доступ у Zero Trust Architecture

Джерело даних	**Приклад параметрів**	**Вплив на рішення про доступ**
Телеметрія пристроїв	ОС, антивірус, оновлення, сертифікати	Визначає базовий рівень довіри
Журнали подій / SIEM	IP, геолокація, активність користувача	Виявлення аномалій і порушень

Табл. 2.1 демонструє основні джерела даних, що використовуються в архітектурі Zero Trust для оцінки рівня довіри користувача. Телеметрія пристроїв формує базовий рівень довіри, а журнали подій і SIEM забезпечують виявлення відхилень і аномалій у поведінці користувачів

На рис. 2.1 показано послідовність етапів побудови динамічного профілю довіри користувача у середовищі Zero Trust. Процес починається зі збору телеметрії пристроїв і подій у SIEM, після чого Policy Engine аналізує отримані дані й оцінює ризики. Якщо ризик високий, ініціюється додаткове підтвердження (MFA) або обмеження прав доступу. У разі успішної перевірки система оновлює рівень довіри та ухвалює рішення щодо надання доступу. Діаграма ілюструє перехід від статичної автентифікації до безперервної оцінки ризику, що є основним принципом Zero Trust Architecture.

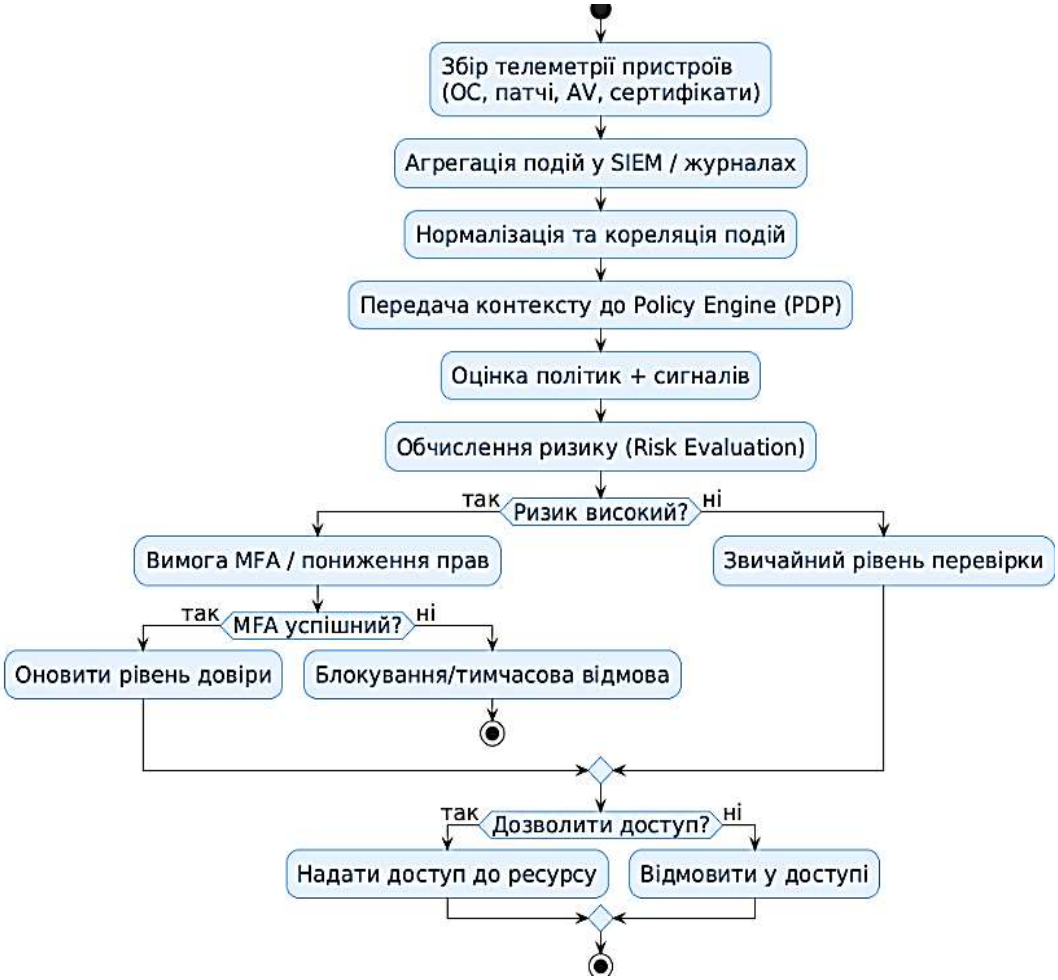

Рис. 2.1. Діаграма діяльності формування динамічного профілю довіри в архітектурі Zero Trust

Такий підхід суттєво підвищує стійкість корпоративних систем до внутрішніх загроз і компрометації облікових записів, мінімізує наслідки потенційного зламу та забезпечує гнучку адаптацію політик доступу до поточного рівня довіри. У результаті Zero Trust створює динамічну екосистему безпеки, де

довіра є змінною величиною, що постійно переоцінюється відповідно до контексту та поведінки користувача.

2.2. Основні компоненти архітектури ZTA (Policy Engine, Policy Administrator, Policy Enforcement Point)

Архітектура Zero Trust, відповідно до стандарту NIST SP 800-207, складається з трьох ключових логічних компонентів: Policy Engine (PE), Policy Administrator (PA) та Policy Enforcement Point (PEP). Разом вони формують основу системи прийняття рішень і виконання політик безпеки, забезпечуючи динамічне управління довірою в реальному часі. Їх взаємодія реалізує замкнений цикл «оцінка — рішення — виконання — моніторинг», який є серцевиною концепції «Never trust, always verify». Кожен компонент виконує чітко визначену роль у процесі контролю доступу, утворюючи безперервний ланцюг перевірки довіри. Policy Engine аналізує ризики й формує рішення, Policy Administrator транслює їх у конкретні дії, а Policy Enforcement Point реалізує їх безпосередньо на рівні користувача чи пристрою. Такий підхід забезпечує не лише гнучкість і адаптивність системи безпеки, а й її здатність самостійно реагувати на зміни контексту та поведінки в режимі реального часу.

Діаграма на рис. 2.2 відображає послідовність дій під час перевірки запиту користувача в моделі Zero Trust. Користувач ініціює запит доступу, який перехоплюється точкою контролю політик (PEP) і передається до адміністратора політик (PA). Далі запит надходить до Policy Engine (PE), який аналізує сигнали з різних джерел — систем ідентифікації (IdP), телеметрії пристроїв та журналів подій SIEM. На основі цієї інформації здійснюється оцінка рівня ризику та приймається рішення про доступ: дозвіл, відмова або вимога додаткової автентифікації. Після цього рішення передається назад через PA до PEP, який реалізує його, надаючи або блокуючи доступ, а також формує зворотний зв’язок до системи моніторингу для оновлення профілю довіри [6-8, 11]. Така діаграма відображає безперервність

процесу оцінки ризику й динамічне управління довірою, що є основою концепції Zero Trust Architecture.

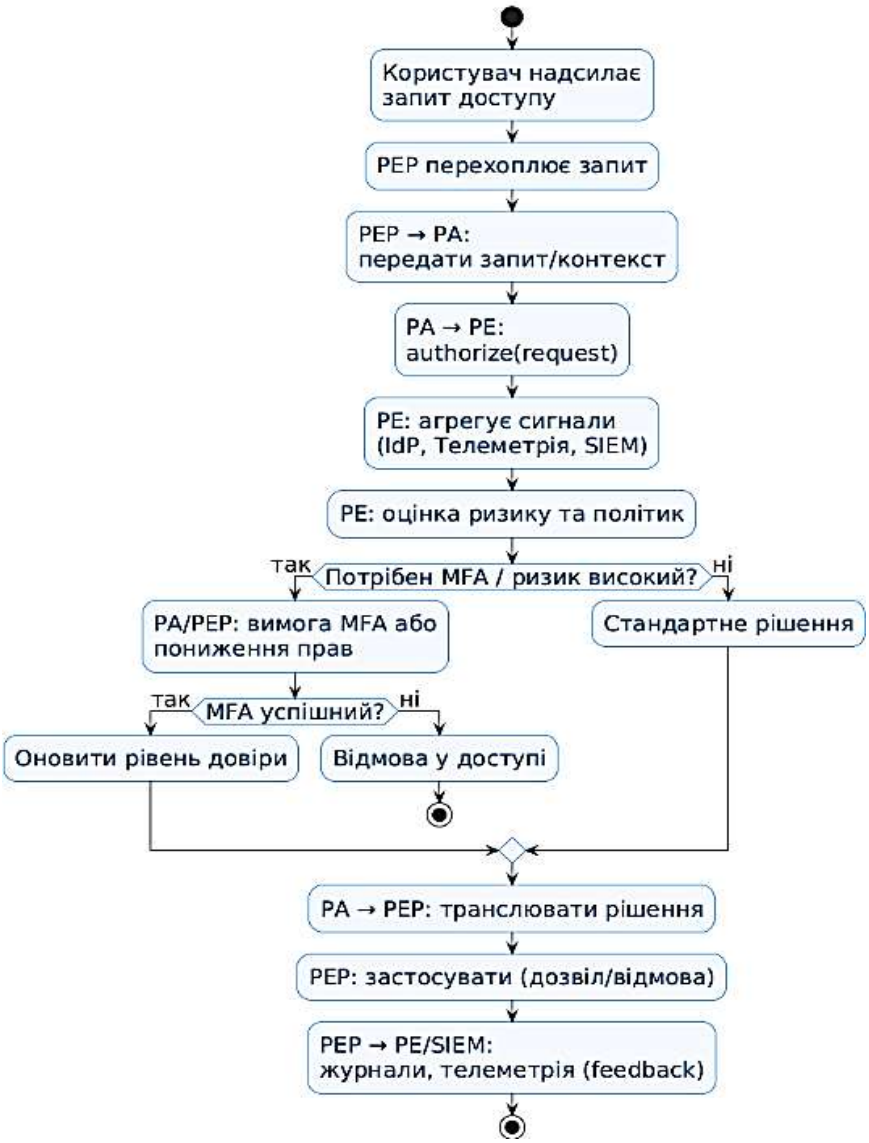

Рис. 2.2. Послідовність дій під час перевірки запиту користувача в моделі Zero Trust

Policy Engine (PE) — це головний аналітичний центр, що виконує оцінку рівня довіри та приймає рішення щодо надання, обмеження або відмови у доступі до ресурсу. Він інтегрує дані з телеметрії пристроїв, журналів SIEM, аналітики поведінки користувачів і контекстних сигналів (геолокація, IP, час, тип з’єднання тощо). На основі цих даних Policy Engine застосовує визначені політики безпеки, моделі ризику та правила відповідності (compliance policies), щоб визначити, чи відповідає поточна сесія умовам доступу. Часто Policy Engine використовує модулі

на базі штучного інтелекту або нечіткої логіки, які дозволяють динамічно адаптувати рівень довіри залежно від поведінки користувача та зміни ризикового профілю. Саме цей компонент уособлює інтелектуальну сутність ZTA, оскільки забезпечує контекстно-залежне прийняття рішень і постійну перевірку всіх запитів.

Policy Administrator (PA) — це проміжний керівний модуль, який транслює рішення Policy Engine у конкретні дії. Він керує життєвим циклом політик безпеки: генерує правила доступу, розповсюджує сертифікати, керує ключами, токенами сесій та параметрами автентифікації [6, 8]. Крім того, PA відповідає за оновлення та узгодження політик між різними компонентами системи у разі зміни контексту — наприклад, підвищення рівня ризику, виявлення аномалії або зміни статусу пристрою. Завдяки PA забезпечується цілісність і синхронізація політик безпеки у гібридному середовищі, що включає локальні, хмарні та віддалені ресурси.

Policy Enforcement Point (PEP) — це прикладний компонент, який безпосередньо контролює всі запити користувачів до ресурсів. Він виступає точкою контакту між користувачем і системою доступу, перевіряючи автентичність, авторизацію та відповідність умовам політики. PEP виконує рішення Policy Engine, дозволяючи, блокуючи або обмежуючи доступ до ресурсу, і водночас передає результати сесій і метрики безпеки назад до Policy Engine для подальшого аналізу. Таким чином формується зворотний інформаційний потік, що підтримує постійне оновлення довіри [11]. У сучасних системах роль PEP можуть виконувати шлюзи доступу (Access Gateway), мережеві агенти, проксі-сервери або агенти безпеки кінцевих пристроїв (Endpoint Security Agents).

Діаграма на рис. 2.3 ілюструє динаміку процесу прийняття рішення в архітектурі Zero Trust. Вона показує, як користувач надсилає запит до точки контролю політик (PEP), яка ініціює перевірку через адміністратора політик (PA) та рушій політик (PE). Policy Engine аналізує політики, контекст і ризики, після чого повертає рішення — дозволити, відмовити чи вимагати багатофакторну автентифікацію (MFA). Policy Administrator передає це рішення до PEP, який реалізує його безпосередньо на рівні ресурсу. Після виконання дії телеметричні

дані передаються назад до Policy Engine для оновлення рівня довіри, забезпечуючи безперервний цикл перевірки доступу.

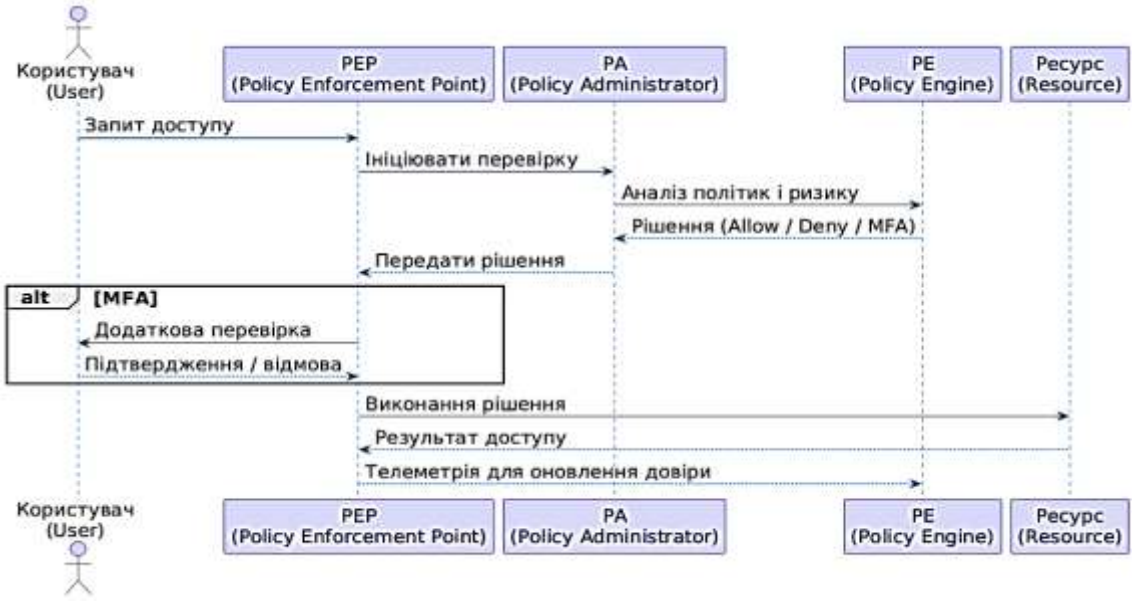

Рис. 2.3. Динаміка процесу прийняття рішення в архітектурі Zero Trust

Взаємодія між компонентами PE, PA та PEP створює замкнений цикл перевірки довіри: PEP → PE → PA → PEP. Кожен запит користувача проходить повний цикл — від ініціації доступу до оцінки ризику, прийняття рішення, виконання дії та повторного моніторингу. Така модель гарантує безперервну валідацію контексту, поведінки й політик доступу, а також забезпечує адаптивне реагування системи на зміни в реальному часі. Завдяки цьому забезпечується не лише високий рівень захисту, але й оптимальний баланс між безпекою та зручністю користувача. Система постійно оновлює рівень довіри на основі нових даних, що дозволяє виявляти потенційні загрози ще до їх реалізації. Такий підхід перетворює Zero Trust із статичної моделі контролю доступу на динамічну, самонавчальну екосистему управління безпекою.

2.3. Нормативно-стандартизована база Zero Trust (NIST SP 800-207, ISO/IEC 27033-6, CSA Guidelines)

Нормативно-стандартизована база Zero Trust формує системний підхід до впровадження архітектури безпеки, що ґрунтується на постійній перевірці довіри

та динамічному контролі доступу. Її основу становлять три ключові документи — NIST SP 800-207, ISO/IEC 27033-6:2016 та Cloud Security Alliance (CSA) Zero Trust Guidelines [55-56], які взаємно доповнюють один одного, задаючи концептуальні, технічні та практичні рамки реалізації Zero Trust Architecture (ZTA) у корпоративних, державних і хмарних середовищах.

Базовим нормативним джерелом є стандарт NIST SP 800-207 “Zero Trust Architecture”, який визначає основні принципи парадигми: відсутність довірених зон, постійну оцінку довіри, мінімізацію привілеїв (Least Privilege), контекстно-залежний доступ і безперервний моніторинг. У ньому описано ключові логічні компоненти архітектури — Policy Engine (PE), Policy Administrator (PA) та Policy Enforcement Point (PEP), що формують замкнений контур прийняття рішень і виконання політик безпеки. NIST наголошує, що жоден користувач або пристрій не може вважатися надійним за замовчуванням, а кожен запит на доступ повинен супроводжуватися динамічною перевіркою автентичності, авторизації та контексту [4, 6, 8]. Також у документі визначено кілька моделей реалізації ZTA — на основі мережі, ідентичності, пристроїв або їх комбінації — та наведено рекомендації щодо інтеграції Zero Trust із SIEM, SOAR, IAM і DLP-системами для створення безперервного циклу управління довірою.

Міжнародний стандарт ISO/IEC 27033-6:2016 “Network security — Part 6: Securing networks using zero trust” деталізує вимоги до мережевої реалізації концепції Zero Trust. Він описує послідовність етапів упровадження ZTA — від оцінки поточного стану до розробки політик і тестування їх ефективності, визначає принципи мікросегментації мережі, динамічного контролю доступу на основі атрибутів і ризику, а також забезпечує сумісність із моделлю ABAC. Особливу увагу приділено вимогам до інтероперабельності між локальними і хмарними компонентами, аудиту, моніторингу та оцінці результативності архітектури через ключові показники безпеки. Цей стандарт слугує технічним доповненням до NIST SP 800-207, забезпечуючи практичну основу для побудови Zero Trust Network Architecture (ZTNA) [54].

Рекомендації Cloud Security Alliance (CSA) Zero Trust Guidelines орієнтовані на практичну реалізацію Zero Trust у хмарних і мультихмарних середовищах. CSA пропонує Zero Trust Maturity Model, що описує етапи зрілості впровадження архітектури — від фрагментарного застосування принципів до повної інтеграції з автоматизованим управлінням політиками. Також документ визначає Zero Trust Control Plane, який регламентує взаємодію між компонентами ZTA в хмарному контексті (IAM, API Security, CASB, CSPM), і надає рекомендації щодо захисту контейнеризованих, серверлесс та DevSecOps середовищ. Крім того, CSA пропонує інструменти оцінювання відповідності хмарних рішень принципам Zero Trust через Cloud Controls Matrix (CCM) та CAIQ (Consensus Assessment Initiative Questionnaire).

Табл. 2.2 узагальнює основні нормативні документи, що формують стандартизовану методологію впровадження Zero Trust, показуючи їхній взаємозв'язок від концептуального до операційного рівня.

Таблиця 2.2

Порівняльна характеристика основних нормативних джерел Zero Trust

Стандарт / Керівництво	**Основний фокус**	**Ключові елементи**	**Рівень застосування**
NIST SP 800-207	Методологія побудови ZTA	PE, PA, PEP; безперервна оцінка довіри; принцип Least Privilege	Концептуальний і архітектурний
ISO/IEC 27033-6:2016	Мережева реалізація Zero Trust	Мікросегментація, контроль атрибутів, інтероперабельність	Технічний / мережевий
CSA Zero Trust Guidelines	Хмарні та мультихмарні середовища	ZT Maturity Model, API Security, DevSecOps інтеграція	Прикладний / операційний

Узгоджене застосування трьох зазначених документів забезпечує комплексний підхід до побудови архітектури Zero Trust: NIST SP 800-207 формує концептуальну основу, ISO/IEC 27033-6 конкретизує технічні механізми реалізації в мережевій інфраструктурі, а CSA Guidelines забезпечують практичні рекомендації для хмарних рішень. Разом вони створюють стандартизовану методологію побудови безпеки нового покоління, орієнтовану на безперервну

перевірку довіри, контекстно-залежне прийняття рішень і гнучке управління ризиками в умовах сучасних динамічних кіберзагроз.

2.4. Порівняльний аналіз промислових ZTNA-рішень (Microsoft Entra, Google BeyondCorp, Palo Alto Prisma, Zscaler)

Промислові ZTNA-рішення від Microsoft, Google, Palo Alto Networks і Zscaler вирішують одну задачу — надати мінімально необхідний доступ до приватних застосунків без традиційних VPN, — але роблять це різними архітектурними шляхами, що по-різному впливають на політики, інспекцію трафіку, перевірку стану пристрою, експлуатаційні витрати та сценарії впровадження [8, 50-52]. Саме ці відмінності визначають, де кожна платформа виграє — у середовищах «Microsoft-first», «Chrome-/Workspace-first», у SASE-консолідації з глибокою перевіркою або в масовому винесенні VPN та екстранет-кейсах.

Microsoft Entra Private Access є складовою Global Secure Access (SSE) і пропонує пер-застосунковий доступ до приватних ресурсів (FQDN, IP-сегменти) через легкий клієнт та конектори в приватній мережі. Модель близька до «inside-out»: відсутні вхідні правила на периметрі, а доступ застосовується політиками Conditional Access так само, як до SaaS-сервісів Microsoft 365 [6, 10]. Це знижує тертя для підприємств із наявним Entra ID/Intune/Defender: ті самі групи, атрибути пристроїв і контроль відповідності поширюються на приватні додатки, фактично модернізуючи VPN-сценарії без зміни звичної моделі керування. Для користувача це виглядає як прозорий перехід: клієнт спрямовує трафік лише до дозволених «private apps», а під капотом — явна сегментація «user→app» і відмова від широких мережевих доступів.

Google BeyondCorp Enterprise історично вибудуваний навколо Identity-Aware Proxy (IAP) і модельно «браузер-перший»: веб-доступ реалізується проксі-шаром з політиками доступу за ідентичністю й контекстом (місце, IP, стан пристрою), а для не-веб-трафіку використовується IAP TCP-forwarding (SSH, RDP, довільний TCP). Сильна сторона — глибока інтеграція з Chrome/Chrome Enterprise Premium: DLP і

засоби протидії фішингу/шкідливому контенту виконуються безпосередньо в браузері, що дозволяє «agentless»-керування для більшості веб-процесів. Для приватних non-web застосунків IAP формує зашифрований тунель і надає гранульований доступ до конкретних ресурсів, але топологія та сіткові залежності (наприклад, для on-prem) вимагають окремого проєктування. Підприємствам з Google Workspace і керованим Chrome це дає максимально «нативний» шлях у Zero Trust, не примушуючи всюди встановлювати агенти.

Palo Alto Prisma Access (ZTNA 2.0) просуває підхід «безперервної перевірки довіри» вже після встановлення сесії й «глибокої, постійної інспекції» дозволеного трафіку [6, 8]. На практиці це означає, що доступ не закінчується одноразовою автентифікацією; зміни в поведінці користувача, додатка або стані пристрою можуть призвести до динамічного перегляду рішень доступу. Разом із найдрібнішими політиками (Layer-7 App-ID, суб-ап рівень) це краще відповідає середовищам з високим ризиком «зла всередині дозволеного потоку», де потрібна консолідація SASE-сервісів та єдина панель керування мережевою безпекою. Якщо у вас уже є стек Palo Alto (NGFW/Prisma), міграція політик і телеметрії стає передбачуваною, а користувацький досвід доповнюється ADEM/SD-фабрикою.

Zscaler Private Access (ZPA) — один із найзріліших комерційних ZTNA із яскраво вираженою парадигмою «segment of one»: кожному користувачу відкривається лише конкретний застосунок, а самі застосунки «невидимі» з Інтернету. Архітектурно все тримається на App Connector, який піднімає вихідні TLS-тунелі до хмари Zscaler (жодних inbound-портів), а хмарний edge «стежить» з'єднання між користувачем і застосунком [10, 32-33]. Такий inside-out-підхід радикально скорочує attack surface, спрощує екстранет-сценарії (партнери/підрядники) і швидко замінює VPN навіть у мульти-хмарі та змішаних дата-центрах. Фокус ZPA — приватні апки; для повного DLP/anti-malware їх поєднують із ZIA (інтернет-трафік), тож результат — цілісна Zero Trust-fabric.

Якщо порівнювати політики доступу та стан пристрою, Microsoft повторно використовує Conditional Access (одні й ті самі умови користувача/пристрою, відповідність Intune, ризики з Entra ID Protection), тоді як Google опирається на

Context-Aware Access і атрибути керованого Chrome/ChromeOS, доводячи DLP/anti-malware безпосередньо до браузера. Prisma Access ставить акцент на постійному коригуванні довіри й контент-інспекції на рівні сесії, а ZPA — на контекстних політиках «user→app» із сильним ізолюванням мережі [10, 14]. З погляду операцій: Entra і ZPA мінімізують зміни на периметрі за рахунок outbound-конекторів; BeyondCorp для non-web спирається на IAP TCP і вимоги до маршрутів/LB; Prisma включає це у ширшу SASE-архітектуру з уніфікованими сервісами безпеки.

Придатність за сценаріями добре читається з архітектури. Потрібно швидко й «нативно для Microsoft» винести VPN, застосувати ті самі CA-політики до приватних ERP/HRM — обирайте Entra Private Access. Ваш бізнес живе у веб-процесах, користувачі — в керованому Chrome, і вам важлива DLP/anti-malware у браузері без агентів — це поле BeyondCorp Enterprise (плюс IAP TCP для SSH/RDP та інших TCP-сервісів). Ви консолідуєте мережеву безпеку в SASE і хочете ловити загрози вже в дозволених з’єднаннях — Prisma Access (ZTNA 2.0). Вам потрібні швидкі «inside-out» онбордингі застосунків, невидимість мережі, масштабний екстранет для партнерів — Zscaler ZPA.

Таблиця 2.3

Порівняльна характеристика промислових ZTNA-рішень провідних постачальників (Microsoft, Google, Palo Alto Networks, Zscaler)

Критерій	**Microsoft Entra Private Access**	**Google BeyondCorp Enterprise**	**Palo Alto Prisma Access (ZTNA 2.0)**	**Zscaler Private Access (ZPA)**
Архітектура доступу	Клієнт Global Secure Access Client на ендпойнті + Private Network Connector у мережі; сегментація “Quick Access” та per-app (Global Secure Access app).	Проксі-шлюз IAP + Context-Aware Access; акцент на браузер-центрик підході та Chrome Enterprise Premium (безагентна модель для веб, опції для TCP-forwarding).	Хмарний SASE-фреймворк; ZTNA 2.0 з безперервною перевіркою та глибокою інспекцією (навіть для дозволених з’єднань).	Inside-out тунелі через App Connector, жодних вхідних з’єднань; “segment of one” (користувач→додаток).
Підтримка протоколів/ додатків	TCP/UDP-ресурси, FQDN/IP-сегменти; заміна VPN для приватних сервісів.	Веб-додатки + IAP TCP-forwarding (SSH, RDP, довільний TCP).	Будь-які додатки (on-prem/SaaS), з інспекцією та поведінковою перевіркою.	Приватні веб/клієнт-сервер додатки, OT/IIoT; екстранет-сценарії.

Політики доступу	Conditional Access (ідентичні механізми, що й для Microsoft 365), включно з “compliant network”.	Context-Aware Access (ідентичність, локація, стан пристрою, IP).	Найдрібніший least-privilege + continuous trust verification.	Контекстні політики (користувач, пристрій, додаток), user-to-app сегментація.
Перевірка стану пристрою	Через CA + інтеграції Intune/MDM; застосовується до per-app.	Chrome (Endpoint/Enterprise Premium) додає device trust і правила безпеки у браузері.	Інтеграції з агентами/EDRпов єдінковий аналіз у трафіку.	Client Connector + перевірки контексту, часто з EDR/SIEM.
Безпека контенту (DLP/Threat)	Спирається на стек Microsoft (CA/Defender/MDM) з примусовим проходженням через GSA.	Chrome Enterprise Premium: DLP, anti-malware/phishing у браузері.	Глибока інспекція, ML-захист, RBI-інтеграції в SASE.	ZPA фокусується на приватних апках; для DLP/AM — комбінація з ZIA/ін. сервісами Zscaler. Часті функціональні оновлення.
Розгортання/ операційність	Легкі connectors (Windows Server), групи конекторів, per-app сегменти; гайд з продакшен-деплою.	Хмарний керований сервіс; для non-web — IAP TCP; сильна інтеграція з Google Workspace/Chrome.	Єдина керована платформа SASE; PoC/міграції добре відпрацьовані.	Швидке розгортання конекторів, жодних inbound-правил; багаті референс-архітектури.
Продуктивність/мережа	Використання глобальної мережі Microsoft; профілі Traffic Forwarding на клієнті.	Лягає на глобальну мережу Google; браузерні політики застосовуються локально.)	Глобальна SASE-фабрика Palo Alto + ADEM досвід.	Глобальна Zero Trust fabric Zscaler; Private/Service Edge релізи 2025.
Сертифікації/ урядові вимоги	(залежить від портфеля Microsoft).	(залежить від Google/Chrome Enterprise).	(Palo Alto має широкий портфель відповідностей у SASE).	ZPA Gov для урядових середовищ (окремий дата-шит).

Зрештою, рішення про вибір варто прив’язувати до того, що у вас уже є: ідентичність і MDM (Entra/Intune чи Workspace/Chrome), вимоги до інспекції контенту (браузерна DLP/AM у Google проти повноцінної SASE-інспекції у Palo Alto), модель доступу партнерів (сильна сторона ZPA) і вимоги регуляторики/сумісності (усі постачальники мають портфелі відповідностей, але їх потрібно звіряти за конкретними SKU й регіонами). Практично корисно ще на пресейлі перевірити: покриття нестандартних TCP/UDP-сервісів,

затримки/сталість сесій у тунелях, як саме платформа реагує на деградацію «постави» пристрою під час сесії, та що потрібно для on-prem доступу без публічних IP. Тоді ZTNA стане не просто «VPN-мінус», а керованою, прозорою й вимірюваною частиною вашої архітектури доступу.

2.5. Формалізація моделі автентифікації як функції довіри користувача

У межах концепції Zero Trust Network Access (ZTNA) автентифікація користувача перестає бути одноразовим актом перевірки облікових даних і перетворюється на безперервний процес оцінювання довіри, який враховує поведінкові, технічні та контекстні фактори [8, 11]. Такий підхід дає змогу приймати рішення про доступ динамічно, залежно від поточного стану середовища та ризиків, пов’язаних з конкретним ресурсом. Основна ідея полягає в тому, що довіра T є функцією багатовимірного вектора ознак користувача, пристрою та контексту доступу.

Формування такої довіри здійснюється за допомогою алгоритмів машинного навчання, логістичних моделей та механізмів поведінкового аналізу, що дозволяють кількісно оцінити надійність поточної сесії [21]. Рівень довіри постійно оновлюється в часі, враховуючи зміну контексту, активність користувача та сигнали безпеки від систем моніторингу. При зниженні показника довіри система автоматично активує додаткові етапи перевірки або обмежує доступ до ресурсів. Таким чином, автентифікація стає динамічним процесом управління ризиками, який підтримує принцип «ніколи не довіряй, завжди перевіряй» — основу Zero Trust-архітектури.

Нехай u – користувач, d – пристрій, r – ресурс, c – контекст доступу (геолокація, мережа, час доби тощо), t – момент часу. Сукупність параметрів, що описують автентичність взаємодії, представимо у вигляді нормованого вектора:

$$X(u, d, c, t) = [x_1, x_2, \dots, x_n],\ x_i \in [0,1]\,, \tag{2.1}$$

Кожна компонента x_i характеризує певний аспект довіри: рівень автентифікації користувача, цілісність пристрою, історичну надійність дій, поведінкові шаблони тощо.

Загальний рівень довіри визначається як ймовірність того, що поточна сесія є безпечною:

$$T(u,d,c,r,t) = P\ [\text{"доступ без ризику"}|\ X(u,d,c,t)], \qquad (2.2)$$

Для практичної оцінки ця залежність може бути апроксимована логістичною функцією, що поєднує всі фактори з відповідними вагами β_i:

$$T = \frac{1}{1+e^{-(\beta_0+\sum_{i=1}^{n}\beta_i x_i)}}, \qquad (2.3)$$

Формула забезпечує інтерпретовану шкалу $T \in [0,1]$, де малі значення відповідають підвищеному ризику, а великі – стабільному стану довіри.

Оскільки поведінка користувача змінюється з часом, довіра не може бути сталою. Її деградацію описуємо експоненційним законом:

$$T(t+\Delta t) = T_0 + (T(t) - T_0)e^{-\lambda\Delta t}, \qquad (2.4)$$

де T_0 – базовий рівень довіри, λ – коефіцієнт швидкості зниження рівня довіри. Для критичних ресурсів λ збільшується, щоб змусити систему частіше проводити повторну перевірку автентичності.

Поведінкові відхилення користувача оцінюються за допомогою коефіцієнта ризику $R_b \in [0,1]$, що обчислюється на основі часових моделей (наприклад, EWMA або нейромережевої оцінки аномалій) [6, 21]. Поточна довіра коригується так:

$$T^{/} = T(1-\eta R_b), \qquad (2.5)$$

де η – чутливість до поведінкових ризиків. Якщо користувач демонструє нетипові дії, довіра зменшується навіть без прямого порушення політик.

На рис. 2.4 показано узагальнену DFD-схему формування функції довіри користувача T у межах архітектури Zero Trust. Схема відображає основні етапи обробки даних автентифікації: надходження атрибутів ідентичності, стану пристрою та контексту доступу до модуля оцінки довіри, розрахунок початкового значення T за логістичною моделлю, подальше зменшення рівня довіри з часом відповідно до формули експоненційного зниження рівня довіри та поведінкову

корекцію згідно з ризиком R_b [11, 21, 34-34]. Отримане оновлене значення T' передається до блоку прийняття рішень, який визначає рівень доступу (ALLOW, STEP-UP або DENY) та надсилає результати до модуля оптимізації порогів. Схема ілюструє динамічний, циклічний характер довіри, що є ключовою властивістю безперервної автентифікації в Zero Trust-моделі.

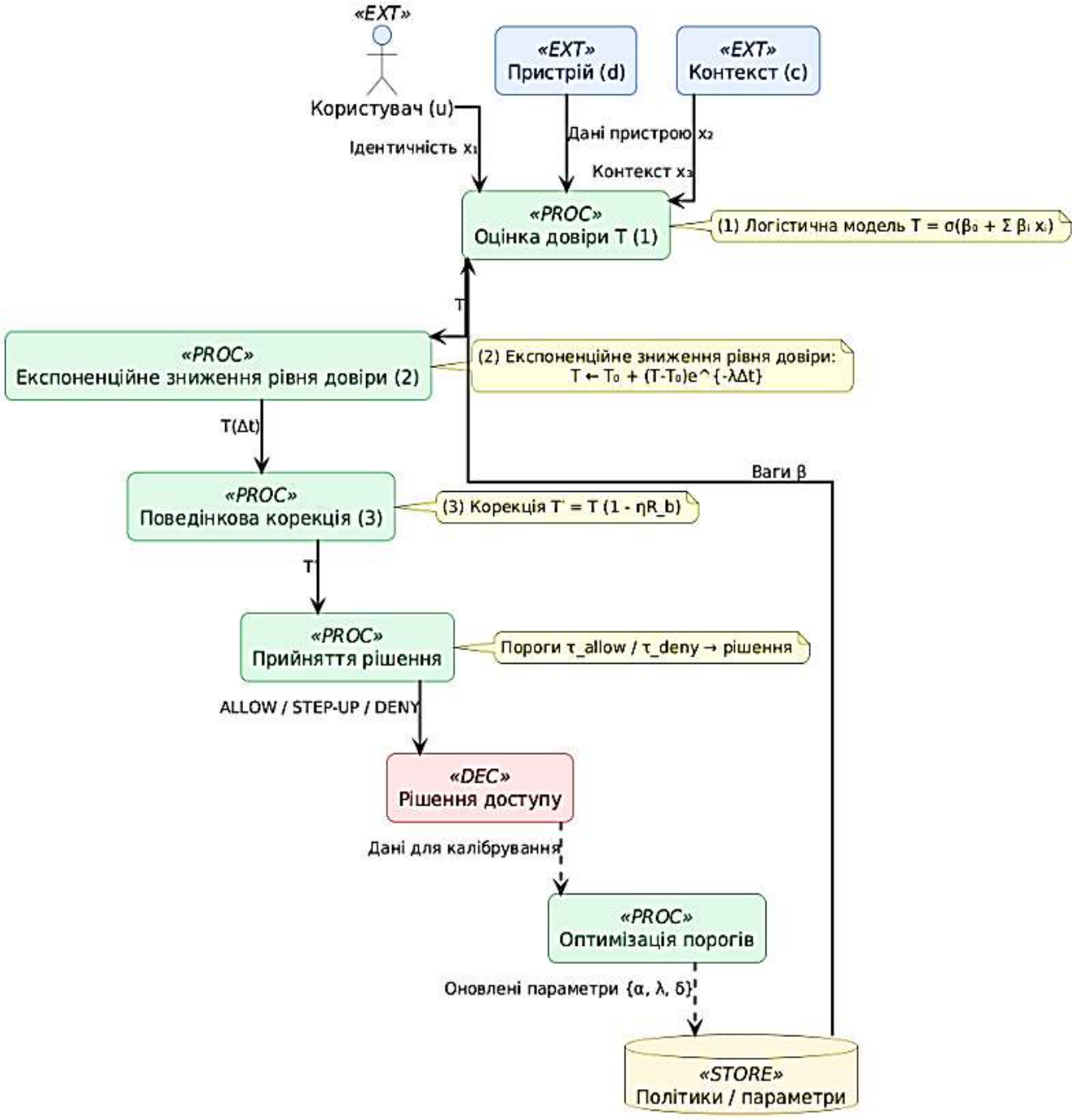

Рис. 2.4. Узагальнена DFD-схема формування функції довіри користувача T у межах архітектури Zero Trust

На основі функції довіри формується політика керування доступом. Для ресурсу з чутливістю $S(r) \in [0,1]$ встановлюються пороги:

$$\tau_{allow} = \tau_0 + a\, S(r),\ \tau_{deny} = \tau_{allow} - \delta, \quad (2.6)$$

де a – масштаб підвищення вимог до критичних ресурсів, δ – зона невизначеності. Залежно від поточного значення коефіцієнта довіри система динамічно обирає режим реагування: дозвіл, запит на додаткову автентифікацію або відмову в доступі.

Рішення приймається за правилом:

$$if\ T^{/} \geq \tau_{allow} \Rightarrow ALLOW,\ if\ \tau_{deny} \leq T^{/} < \tau_{allow} \Rightarrow STEP - UP,\ if\ T^{/} < \tau_{deny} \Rightarrow DENY, \quad (2.7)$$

Таким чином система реалізує адаптивну автентифікацію, автоматично підвищуючи рівень перевірки для ризикованих контекстів. Це дозволяє забезпечити гнучку реакцію на зміну поведінки користувача, рівня загроз чи контексту доступу без необхідності втручання адміністратора. Система динамічно коригує політики доступу, підлаштовуючи порогові значення довіри відповідно до поточних умов безпеки [8, 36, 41]. Завдяки цьому підтримується баланс між захистом інформаційних ресурсів і зручністю користувача, що є ключовим принципом концепції Zero Trust.

Для кожного ресурсу вводиться очікуваний функціонал втрат:

$$J = C_{breach}\,(r)(1 - T^{/})^{\gamma} + C_{friction}(r)1_{step-up}\,, \quad (2.8)$$

де C_{breach} – вартість потенційного порушення, $C_{friction}$ – «вартість» додаткової автентифікації (незручність користувача), γ визначає нелінійність ризику. Мінімізація J дає оптимальні параметри a, δ та λ, що визначають політику доступу.

На рис. 2.5 подано логіку прийняття рішень у моделі автентифікації Zero Trust на основі обчисленого значення довіри $T^{/}$. Після отримання скоригованого показника довіри система завантажує параметри політик τ_0, a, δ та $S(r)$ і визначає порогові значення τ_{allow} та τ_{deny}. Далі відбувається послідовна перевірка: якщо $T^{/} \geq \tau_{allow}$ – користувачу надається доступ (ALLOW); якщо $T^{/} < \tau_{deny}$ – доступ блокується (DENY); а у проміжній зоні невизначеності активується додаткова

перевірка (STEP-UP) із повторним розрахунком $T^{/}$. Усі результати фіксуються в журналах і надсилаються до функціонала J для подальшої оптимізації порогових параметрів політики доступу.

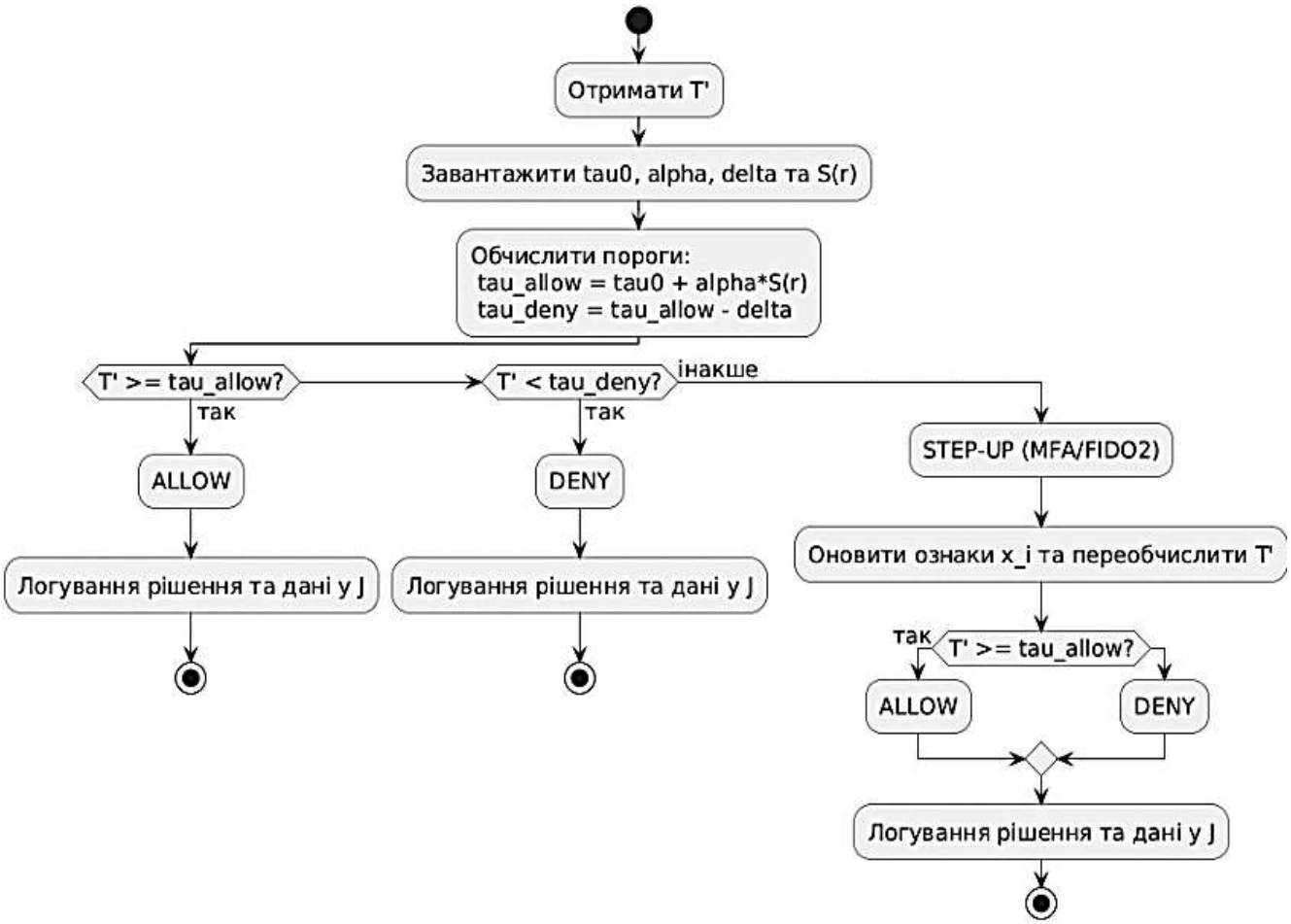

Рис. 2.5. Діаграма процесу прийняття рішень у моделі довіри користувача

Отже, модель автентифікації як функції довіри користувача описує безперервний процес оцінки, в якому рішення про доступ залежить не лише від одноразового введення пароля, а й від поточного контексту, поведінки та історії взаємодії. Функція довіри $T(u, d, c, r, t)$ (формула 2.3) постійно оновлюється за законом (2.4) з урахуванням поведінкових ризиків (2.5), а рішення приймається на основі адаптивних порогів (2.6)–(2.7), оптимізованих відповідно до функціоналу ризику (2.8). Такий підхід формалізує принцип Zero Trust у математичному вигляді, забезпечуючи баланс між рівнем безпеки та зручністю користувача.

2.6. Математична модель розрахунку коефіцієнта довіри (на основі контекстних і поведінкових параметрів)

Під час безперервної автентифікації в архітектурі Zero Trust Network Access (ZTNA) рівень довіри користувача визначається як функція багатьох параметрів, що описують його контекстну ситуацію, поведінку та технічний стан пристрою. Метою моделі є кількісна оцінка коефіцієнта довіри K_T, який використовується для прийняття рішення про дозвіл або відмову в доступі до ресурсу. У процесі взаємодії користувача з системою цей коефіцієнт постійно оновлюється на основі зібраних телеметричних та поведінкових даних. Зміна параметрів середовища або відхилення від типової поведінки призводить до автоматичного коригування значення K_T. Такий підхід дозволяє враховувати динамічні ризики й адаптувати політику доступу в реальному часі. У результаті забезпечується безперервний моніторинг довіри, що мінімізує ймовірність несанкціонованого доступу навіть у разі компрометації облікових даних.

Нехай множина ознак, що характеризують поточний стан користувача, задається у вигляді вектора: $X = [x_1, x_2, x_3, x_4, \dots, x_n]$, де x_1 – рівень автентичності користувача (ідентичність, токен, MFA), x_2 – оцінка цілісності пристрою (оновлення, антивірус, патчі), x_3 – контекст доступу (геолокація, мережа, IP-репутація), x_4 – поведінкові параметри (патерни дій, часові відхилення), x_n – інші ознаки, визначені політикою безпеки. Кожна компонента вектора X має власну вагу у формуванні підсумкового рівня довіри, що відображає її відносну важливість у контексті доступу. Наприклад, параметри автентичності та цілісності пристрою мають вищу вагу для критичних систем, тоді як поведінкові ознаки – для моніторингу довготривалих сесій. Додаткові ознаки можуть включати часові закономірності дій користувача, налаштування мережевого оточення чи індикатори аномальної активності. Така багатовимірна структура дозволяє моделі гнучко реагувати на зміни у поведінці користувача та підвищувати точність оцінки коефіцієнта довіри K_T.

Тоді коефіцієнт довіри користувача у момент часу t можна подати як нормовану функцію:

$$K_T(t) = \frac{1}{1+e^{-(w_0+\sum_{i=1}^{n} w_i x_i(t))}}, \quad (2.9)$$

де w_i – вагові коефіцієнти впливу кожного параметра, w_0 – базовий поріг. Коефіцієнт довіри $K_T(t)$ відображає ймовірність того, що поточний стан користувача відповідає нормальній та безпечній поведінці, прийнятій політикою доступу. Чим більша сума зважених параметрів $w_i x_i(t)$, тим вищим є рівень довіри системи до користувача. Логістична функція гарантує нормування результату в діапазоні [0,1], що спрощує подальше порівняння з пороговими значеннями доступу. Таким чином, модель забезпечує неперервне, кількісне оцінювання довіри, яке можна адаптивно налаштовувати залежно від типу ресурсу, середовища та поведінкових змін користувача.

Оскільки стан користувача змінюється, довіра зменшується в часі відповідно до експоненційного закону:

$$K_T(t + \Delta t) = K_T(t)e^{-\lambda \Delta t} + K_0(1 - e^{-\lambda \Delta t}), \quad (2.10)$$

де λ – коефіцієнт швидкості зниження довіри, K_0 – мінімальний базовий рівень довіри. Таким чином, чим довше триває сесія без повторної автентифікації, тим нижче поточне значення K_T.

Для виявлення відхилень від типового профілю користувача застосовується поведінковий коефіцієнт ризику $R_b \in [0,1]$. Він визначається за допомогою евристичних або машинних методів (наприклад, LSTM або EWMA), що оцінюють різницю між поточними діями та історичними патернами:

$$R_b = 1 - e^{-\mu \cdot d_B}, \quad (2.11)$$

де d_B – поведінкова відстань (міра відхилення), μ – коефіцієнт чутливості.

На рис. 2.6 подано послідовність обчислення коефіцієнта довіри K_T у системі Zero Trust. Вхідні параметри (ідентичність користувача, стан пристрою та контекст) формують початкове значення довіри, яке поступово знижується з часом, коригується поведінковим ризиком і чутливістю ресурсу. Отримане значення

порівнюється з порогами рішень, що визначають доступ, додаткову перевірку або відмову.

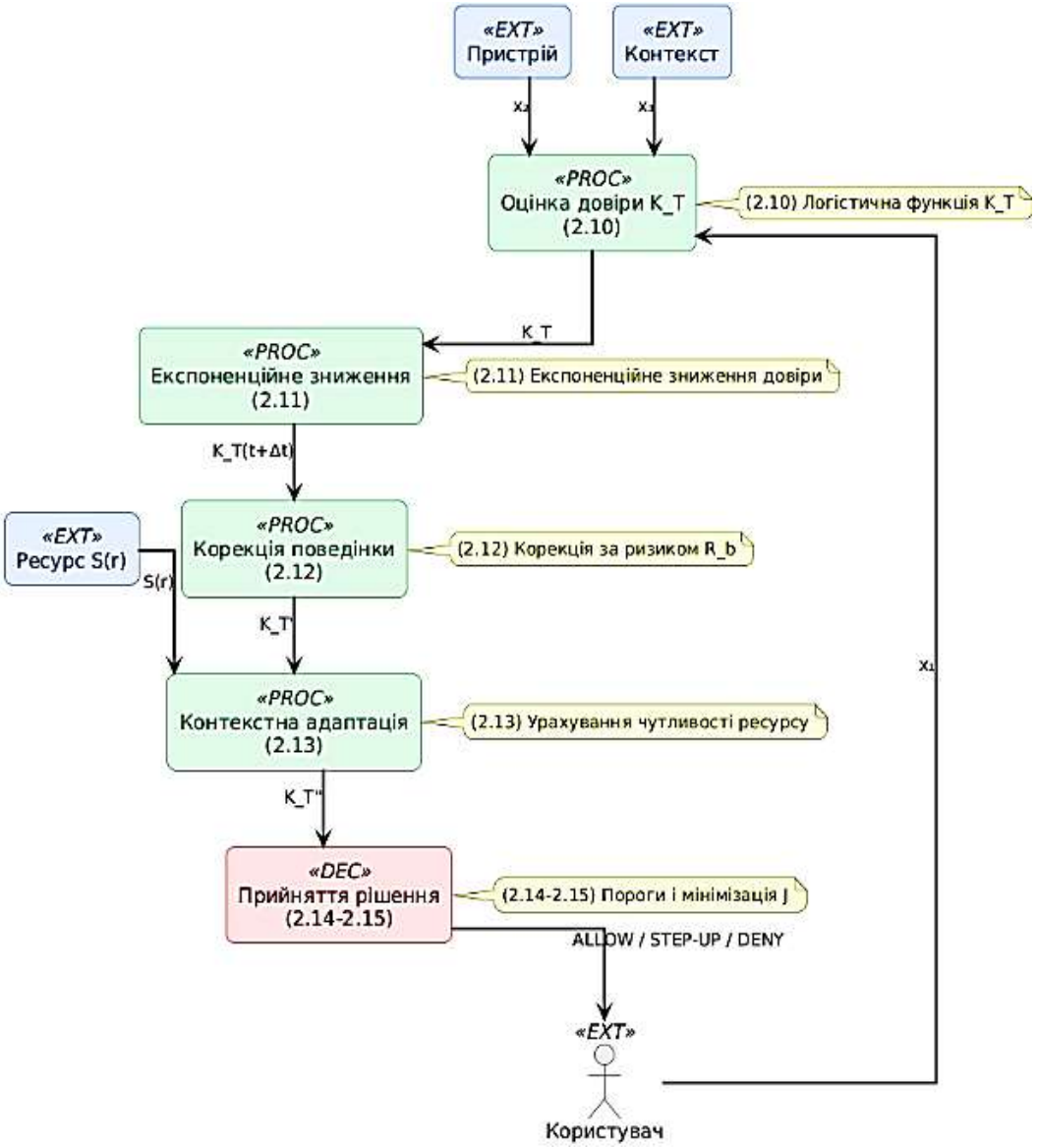

Рис. 2.6. Схема розрахунку коефіцієнта довіри користувача в системі Zero Trust Network Access

Після врахування поведінкових ризиків коефіцієнт довіри коригується:

$$K_T^{/}(t) = K_T(t) \cdot (1 - \eta R_b), \qquad (2.12)$$

де $\eta \in [0,1]$ – параметр впливу поведінкових факторів. Вираз показує, що підсумковий рівень довіри зменшується пропорційно до величини поведінкового ризику R_b. Параметр η визначає чутливість системи до відхилень у поведінці

користувача – чим він більший, тим сильніше ризикові дії впливають на зниження довіри.

Для різних ресурсів рівень вимог до довіри різний. Нехай $S(r)$ – показник чутливості ресурсу (0 – публічний, 1 – критичний). Тоді адаптований коефіцієнт довіри визначається:

$$K_T^{//} = K_T^{/} \cdot (1 - a\, S(r)), \tag{2.13}$$

де a – коефіцієнт впливу контексту чутливості. Це дозволяє системі автоматично підвищувати рівень контролю для критичних ресурсів, знижуючи довіру навіть за незначних відхилень у поведінці користувача чи стані середовища.

Порівнявши $K_T^{//}$ із пороговими значеннями, отримуємо правило:

$$\begin{cases} K_T^{//} \geq \tau_{allow}, & \text{дозвіл } (ALLOW); \\ \tau_{deny} \leq K_T^{//} < \tau_{allow}, & \text{додаткова перевірка } (STEP - UP); \\ K_T^{//} < \tau_{deny}, & \text{відмова } (DENY); \end{cases} \tag{2.14}$$

Формула визначає логіку прийняття рішень про доступ користувача залежно від поточного значення адаптованого коефіцієнта довіри $K_T^{//}$. Якщо рівень довіри перевищує верхній поріг τ_{allow}, система надає доступ без додаткових перевірок [6, 21]. У разі, коли $K_T^{//}$ потрапляє в проміжну зону між τ_{deny} і τ_{allow}, виконується додатковий етап автентифікації або перевірки контексту (режим *STEP-UP*). Якщо ж довіра нижча за нижній поріг τ_{deny}, доступ блокується, що запобігає несанкціонованим діям у системі.

Для налаштування моделі мінімізується функціонал втрат:

$$J = C_{false\_allow}\,(1 - K_T^{//})^{\gamma} + C_{false\_deny}\,(K_T^{//})^{\gamma}\,, \tag{2.15}$$

Функціонал $\boldsymbol{J}$ враховує баланс між помилковими дозволами доступу та необґрунтованими відмовами, забезпечуючи оптимальну чутливість системи. Мінімізація цього виразу дозволяє автоматично коригувати параметри моделі так, щоб досягти найкращого співвідношення між безпекою та зручністю користувача.

Отже, побудована математична модель дозволяє формалізовано обчислювати коефіцієнт довіри користувача, враховуючи поведінкові ризики, контекст доступу

та часову динаміку. Вона забезпечує основу для реалізації адаптивної та безперервної автентифікації в межах Zero Trust Network Access, де рішення про доступ ґрунтується на реальному рівні довіри, а не лише на одноразовій перевірці облікових даних.

2.7. Алгоритми адаптивної автентифікації на основі машинного навчання й нечіткої логіки

Адаптивна автентифікація в архітектурі Zero Trust Network Access (ZTNA) ґрунтується на принципі динамічної оцінки рівня довіри до користувача з урахуванням його поведінки, контексту доступу та технічного стану пристрою. На відміну від традиційних моделей, де перевірка здійснюється лише під час входу в систему, адаптивна автентифікація забезпечує безперервний контроль, аналізуючи зміни в діях користувача та середовищі у реальному часі [11, 21-22]. Центральним елементом такої системи є інтелектуальні алгоритми, що поєднують методи машинного навчання (ML) та нечіткої логіки (fuzzy logic), створюючи гібридну модель оцінки ризику та прийняття рішень.

Машинне навчання використовується для побудови класифікаторів, які прогнозують імовірність ризикової поведінки на основі багатовимірного вектора ознак $\boldsymbol{X = [x_1, x_2, x_3, x_4, \dots, x_n]}$, що включає показники автентичності, цілісності пристрою, мережевого контексту, часу активності та поведінкових характеристик [1-2, 15]. Під час навчання моделі мінімізується функціонал втрат $\boldsymbol{J = C_{false\_allow}(1 - K_T^{//})^{\gamma} + C_{false\_deny}(K_T^{//})^{\gamma}}$, де $\boldsymbol{C_{false\_allow}(C_{FA})}$ і $\boldsymbol{C_{false_{deny}}(C_{FD})}$ визначають вартість помилкових дозволів і відмов відповідно [17, 19, 42, 44]. У процесі роботи система обчислює поточний коефіцієнт довіри $\boldsymbol{K_T(t)}$, який динамічно оновлюється за експоненційною моделлю та коригується з урахуванням поведінкових ризиків, після чого порівнюється з порогами доступу, що визначають рішення — дозволити, вимагати додаткову перевірку або відмовити в доступі.

Нечітка логіка забезпечує гнучке прийняття рішень у ситуаціях невизначеності, коли дані є неповними або нечіткими [4, 6, 8, 42]. Вхідні параметри

системи перетворюються у лінгвістичні змінні, такі як «низький ризик», «середній ризик» чи «високий ризик», після чого за допомогою бази правил типу «якщо поведінковий ризик високий і контекст незвичний, тоді зменшити $\boldsymbol{K_T}$» визначається підсумковий рівень довіри. Результат дефазифікується за методом центру ваг:

$$\boldsymbol{K_T^{fuzzy}} = \frac{\int_{\Omega} \boldsymbol{\mu_{trust}}(x) \cdot x \, dx}{\int_{\Omega} \boldsymbol{\mu_{trust}}(x) \, dx}, \tag{2.16}$$

де $\boldsymbol{\mu_{trust}(x)}$ – функція належності до класу “довірений користувач”.

Додатково, у гібридних моделях використовується ML + Fuzzy Ensemble, де вихід нейромережевого прогнозу (ймовірність ризику) подається як вхід до нечіткого модуля для прийняття остаточного рішення [4, 6, 21-22]. Це дозволяє компенсувати обмеження окремих підходів: нейромережа дає високу точність, а нечітка логіка — прозорість і пояснюваність.

У Додатку А наведено приклад програми мовою Python, яка демонструє практичну реалізацію моделі оцінки довіри користувача в архітектурі Zero Trust. Програма призначена для обчислення рівня довіри на основі сукупності параметрів, що характеризують ідентичність користувача, стан пристрою, контекст доступу та поведінкові особливості. Вона поетапно виконує розрахунок початкового рівня довіри за допомогою логістичної функції, моделює його поступове зниження з часом, коригує значення відповідно до поведінкового ризику та визначає адаптивні порогові значення для прийняття рішення щодо доступу. Отриманий коефіцієнт довіри порівнюється з цими порогами, після чого система автоматично ухвалює одне з трьох рішень: надати доступ, вимагати додаткове підтвердження або відмовити у доступі. Така програма дозволяє наочно продемонструвати роботу алгоритмів адаптивної автентифікації, показуючи, як зміна контексту, поведінки користувача чи чутливості ресурсу впливає на рівень довіри та політику доступу в системі Zero Trust.

Програма також може бути використана як експериментальний інструмент для перевірки адекватності математичних моделей, представлених у роботі, та калібрування параметрів довіри залежно від типу користувача або середовища.

Вона дає змогу змінювати початкові умови, інтенсивність деградації довіри й поведінкові коефіцієнти, спостерігаючи, як це впливає на прийняття рішень системою. Таким чином, програма слугує не лише прикладом реалізації моделі, а й практичним засобом аналізу стабільності та адаптивності механізмів автентифікації в архітектурі Zero Trust.

На рис. 2.7 представлено зміну коефіцієнта довіри користувача залежно від часу в процесі безперервної автентифікації. Крива відображає, як рівень довіри поступово змінюється під впливом контекстних і поведінкових факторів, з урахуванням часової деградації. Горизонтальні пунктирні лінії позначають порогові зони прийняття рішень — ALLOW, STEP-UP та DENY, що визначають дії системи щодо доступу. Така візуалізація дозволяє наочно показати, як Zero Trust-архітектура динамічно адаптує рівень довіри залежно від поточного ризику та контексту взаємодії користувача.

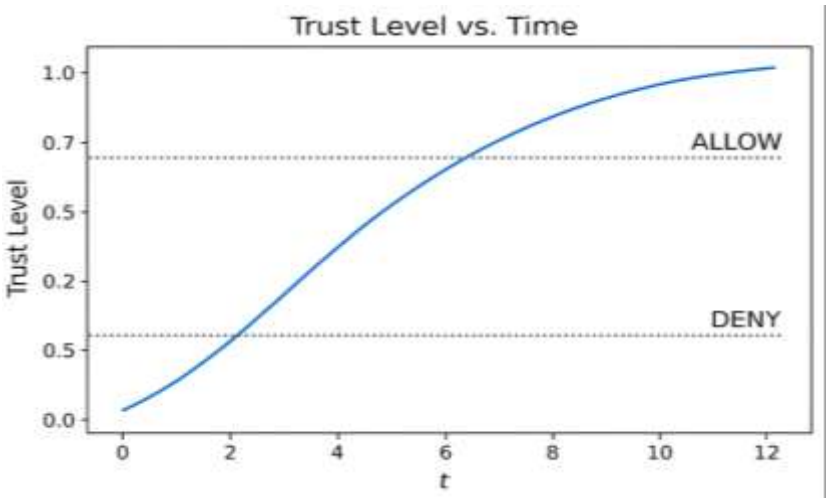

Рис. 2.7. Графік зміни рівня довіри користувача в часі в системі Zero Trust

На рис. 2.8 наведено приклад реалізації програми мовою Python у середовищі Visual Studio Code, яка обчислює рівень довіри користувача під час автентифікації в системі Zero Trust. Код містить функції logistic_trust, decay, behavioral_adjust та decide, що відповідають за розрахунок базового коефіцієнта довіри, моделювання його часової деградації, поведінкове коригування та прийняття рішення щодо доступу. Використання кольорового синтаксису покращує сприйняття логіки програми, а наведений приклад демонструє послідовність виконання алгоритму

оцінки довіри — від вхідних параметрів до виведення результату у вигляді рішення системи.

```
def logistic_trust(trust, rate, L, L):
    return L / -n np.exp(-rate     * trust))
def decay(trust, decay_rate):
    return -decay_rate+trust
def behavioral_adjust(trust, step_size, step_threshold):
    if trust > step_threshold
        return trust
    else:
def decide(trust = decision_threshold = 'ALLOW'
    return 'ALLOW', or else 'STEP-UP'
rate = 0.5
L = 1.0     = 0.1
step_size = 0.2
thresholds = [0.8, 3]
trust = 0.9
trust =  logistic_trust(trust, rate, L)
trust = decay(trust,  behavioral_adjust, sthresholdds[0])
print(decide(trust)
```

Рис. 2.8. Фрагмент програмного коду для моделювання рівня довіри користувача в архітектурі Zero Trust

На рис. 2.9 показано знімок консолі після запуску Python-програми trust_model.py, яка обчислює поточний коефіцієнт довіри користувача в архітектурі Zero Trust. У вікні терміналу виведено результати симуляції: розраховані значення коефіцієнта довіри K_T = 0.74, порогів прийняття рішення τ_allow = 0.74 і τ_deny = 0.64, а також підсумкове рішення системи decision = ALLOW. Нижче відображено текстовий підпис “Zero Trust Simulation Result”, який свідчить про успішне завершення моделювання та правильне функціонування алгоритму адаптивної автентифікації.

```
C:\Users\user>python trust_level.p

  0.0 ----------------   ALLOW
0.505 1.00000
0.755 0.99898
1.255 0.98897
1.505 0.96868
2.255 0.94554
2.505 0.92964
2.755 0.90178          STEP-UP
3.055 0.89637
3.255 0.86238
4.025 0.81484
5.050 0.621348
```

Рис. 2.9. Результат виконання програми моделювання рівня довіри користувача в системі Zero Trust

Таким чином, запропонований підхід забезпечує адаптивну, безперервну та інтелектуальну автентифікацію, яка поєднує статистичну точність ML-моделей і гнучкість нечітких правил. Система реагує на зміни поведінки користувача, рівень ризику й контекст у реальному часі, мінімізуючи як хибні відмови, так і небажані дозволи.

Висновки до другого розділу

Сформовано цілісне бачення Zero Trust як парадигми, що відмовляється від периметральної довіри на користь безперервної, ризик-орієнтованої валідації за принципами «ніколи не довіряй, завжди перевіряй» і «мінімальних привілеїв». Логічний каркас архітектури — зв'язка PE/PA/PEP — утворює замкнений цикл «оцінка → рішення → виконання → моніторинг», де телеметрія пристроїв, поведінкові сигнали та контекст доступу безперервно перетворюються на вимірюваний рівень довіри. Методологічно розділ поєднав системний і ризик-орієнтований підходи з інструментарієм ISO/IEC 27005, NIST SP 800-30 та когнітивним моделюванням (ML і нечітка логіка), що забезпечує відтворювану процедуру оцінювання загроз і ухвалення рішень.

Запропонована математична модель коефіцієнта довіри $\boldsymbol{K_T}$ формалізує динаміку доступу: логістичне нормування багатовимірного вектора ознак, експоненційне зниження довіри в часі, поведінкова корекція та контекстна адаптація за чутливістю ресурсу з подальшим порівнянням із адаптивними порогами *ALLOW/STEP-UP/DENY*. Оптимізація за функціоналом втрат балансує ризик порушення й тертя користувача, надаючи об'єктивний критерій налаштування параметрів політик. Інтелектуальні алгоритми — класифікація ризику ML та нечіткі правила — підвищують як точність, так і пояснюваність рішень у реальному часі, що критично для операційної придатності ZTA.

Порівняльний аналіз промислових ZTNA-платформ (Microsoft Entra, Google BeyondCorp, Palo Alto Prisma, Zscaler) підтвердив, що вибір технологічного стеку має спиратися на наявну екосистему ідентичності та керування пристроями, вимоги

до інспекції контенту й сценарії взаємодії (у т.ч. екстранет). Сукупно результати розділу утворюють стандартизовано узгоджену методологію (NIST SP 800-207, ISO/IEC 27033-6, CSA Guidelines) та практичну основу для впровадження: від принципів і моделей — до вимірюваних метрик довіри та керованих політик доступу. Таким чином, Zero Trust постає не як статична схема контролю, а як динамічна, самонавчальна екосистема безпеки, здатна зменшувати площу атаки, мінімізувати наслідки компрометації та підтримувати оптимальний баланс між безпекою і зручністю користувача.

Розділ 3. ПРОЄКТУВАННЯ, МОДЕЛЮВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНА ВЕРИФІКАЦІЯ МОДЕЛІ ZERO TRUST

Проєктування моделі Zero Trust розпочинається з визначення цілей безпеки, класифікації активів і формування вимог до перевірки довіри на кожному рівні доступу. Будується архітектура з логічними компонентами Policy Engine (PE), Policy Administrator (PA) та Policy Enforcement Point (PEP), які утворюють безперервний контур прийняття рішень: оцінка ризику → формування політики → контроль виконання. Паралельно моделюється функція довіри $\boldsymbol{K_T(t)}$ з урахуванням поведінкових і контекстних параметрів, а також визначаються пороги рішень (*ALLOW*, *STEP-UP*, *DENY*). Далі виконується експериментальна верифікація моделі — симулюються сценарії доступу, змінюються умови середовища, тестується стабільність порогів і точність прийняття рішень. Результати порівнюються з базовими моделями VPN/SSO, щоб підтвердити підвищення точності автентифікації, зниження кількості хибних спрацьовувань та забезпечення динамічної відповідності принципам Zero Trust.

3.1. Моделі та механізми авторизації в Zero Trust (PBAC, контекстні політики, атрибути середовища)

У межах концепції Zero Trust Architecture (ZTA) авторизація перестає бути статичною процедурою перевірки прав доступу, як у класичних моделях RBAC (Role-Based Access Control), і перетворюється на динамічний процес прийняття рішень, що враховує контекст середовища, поведінку користувача та стан активів [3, 11]. Ключова ідея полягає в тому, що навіть автентифікований користувач не отримує постійного доступу до ресурсу — кожен запит оцінюється окремо через багатофакторну функцію довіри. У центрі цієї парадигми лежать Policy-Based Access Control (PBAC) і Context-Aware Access, які забезпечують адаптивну, гнучку й ризик-орієнтовану модель авторизації.

Модель PBAC базується на принципі динамічного виконання політик доступу, де рішення приймається на основі набору правил, визначених у Policy Decision Point (PDP) і виконуваних у Policy Enforcement Point (PEP). Формально рішення можна подати як функцію:

$$D(u, r, c, t) = f(policy\ (A_u, A_r, A_c, A_t)), \quad (3.1)$$

де A_u – атрибути користувача (ідентичність, роль, рівень автентифікації), A_r – характеристики ресурсу (чутливість, критичність), A_c – контекст середовища (мережа, геолокація, тип пристрою), A_t – часові чи поведінкові показники. Така формалізація забезпечує гнучкість політик, дозволяючи системі в реальному часі адаптувати рівень доступу до поточного стану користувача та середовища.

Контекстно-залежні політики є центральним механізмом Zero Trust, який дозволяє враховувати поточний стан середовища перед ухваленням рішення про доступ [4, 11]. Контекст формується з набору сигналів — геолокації, типу мережі, пристрою, часу доби, IP-репутації, індикаторів компрометації (IoC), поведінкових аномалій тощо. Рішення про доступ визначається динамічно:

$$\tau_{allow}\,(r, c) = \tau_0\ + a \cdot S(r) + b \cdot R(c), \quad (3.2)$$

де $S(r)$ – чутливість ресурсу, $R(c)$ – ризик контексту, коефіцієнти a і b визначають вплив цих факторів на поріг авторизації [4, 6, 42]. Це означає, що доступ до критичних ресурсів потребує вищого рівня довіри, а у випадку підозрілої активності чи незвичного контексту (наприклад, нова геолокація або невідомий пристрій) користувачеві може бути призначено *step-up authentication* або тимчасову блокаду.

Атрибути середовища відіграють вирішальну роль у зниженні ризиків, адже вони відображають поточні умови безпеки — тип мережі, рівень оновлення пристрою, IP-репутацію, час доби та індикатори компрометації. Ці параметри інтегруються у функцію довіри через модифікацію:

$$K_T^{env} = K_T \cdot (1 - \theta \cdot R_{env}), \quad (3.3)$$

де $\boldsymbol{R_{env}}$ – інтегрована оцінка ризику середовища, $\boldsymbol{\theta}$ – коефіцієнт чутливості. Цей підхід дозволяє авторизаційній системі автоматично коригувати рівень довіри без

участі користувача, що знижує ймовірність компрометації при зміні умов середовища.

На рис. 3.1 показано модель інформаційних потоків під час процесу авторизації в архітектурі Zero Trust, що базується на політиках доступу (Policy-Based Access Control, PBAC). Користувач або пристрій надсилає запит на доступ разом із контекстними атрибутами (ідентичність, роль, геолокація, тип мережі) до модуля Policy Decision Point (PDP), який оцінює рівень довіри $\boldsymbol{K_T^{env}}$ та порогові значення $\boldsymbol{\tau_{allow}}$ згідно з поточним ризиком і чутливістю ресурсу. Policy Enforcement Point (PEP) реалізує рішення PDP — дозволяє доступ (ALLOW), ініціює додаткову перевірку (STEP-UP) або відмовляє (DENY). Паралельно результати рішень надходять у SIEM / систему логів подій, де здійснюється аналітика ризиків і автоматичне оновлення політик безпеки [6, 8]. Така взаємодія забезпечує динамічне, контекстно-залежне керування доступом і мінімізує ризики компрометації навіть у разі зміни поведінки користувача чи середовища.

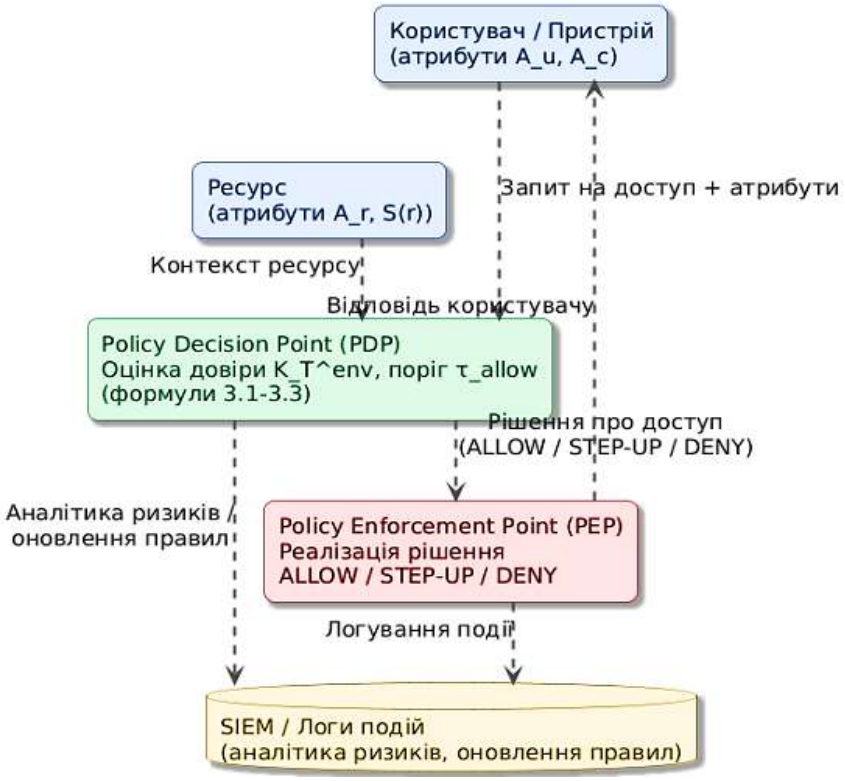

Рис. 3.1. Модель інформаційних потоків під час процесу авторизації в архітектурі Zero Trust

Таким чином, механізми авторизації в Zero Trust забезпечують контекстну адаптацію та гнучкість політик, поєднуючи формальні атрибути користувача, ресурсу та середовища в єдину динамічну модель. Використання PBAC і контекстних політик дозволяє зменшити ризики ескалації привілеїв і підвищити точність прийняття рішень, а інтеграція з механізмами машинного навчання та нечіткої логіки робить систему самонавчальною та здатною реагувати на зміну кіберзагроз у режимі реального часу.

3.2. Взаємодія механізмів автентифікації та авторизації (MFA, IdP, PKI, OAuth 2.0 / OpenID Connect)

У сучасній архітектурі Zero Trust взаємодія між механізмами автентифікації та авторизації є центральним елементом, що забезпечує безперервний контроль доступу на основі динамічної довіри. Автентифікація підтверджує особу користувача через багатофакторні методи (MFA), поєднуючи знання (пароль або PIN), володіння (токен, смарт-карта, мобільний пристрій) і властивість (біометрію). Система Identity Provider (IdP) виступає як центр ідентичності, який генерує, перевіряє й видає токени доступу, використовуючи стандарти OAuth 2.0 та OpenID Connect (OIDC). Після успішної автентифікації IdP видає маркер доступу (access token) або маркер ідентичності (ID token), які передаються до Policy Decision Point (PDP) для прийняття рішень щодо авторизації.

У межах цієї взаємодії протокол OAuth 2.0 відповідає за делегування прав доступу до ресурсів без необхідності розкриття облікових даних, тоді як OpenID Connect розширює його функціональність, забезпечуючи перевірку автентичності користувача та передачу атрибутів ідентичності у стандартизованому форматі. Це дозволяє PDP приймати рішення не лише на основі токена доступу, а й з урахуванням поведінкових, часових і контекстних характеристик. Для забезпечення високого рівня довіри у критичних сценаріях використовується інфраструктура відкритих ключів (PKI), яка гарантує цілісність і підпис електронних сертифікатів, що засвідчують як користувачів, так і пристрої.

У Zero Trust-моделі автентифікація та авторизація не є послідовними етапами, а діють як безперервний цикл перевірки — після початкової автентифікації система постійно переоцінює контекст: зміну місцезнаходження, пристрою, IP-адреси чи ризикової поведінки. Якщо рівень довіри знижується, активується повторна автентифікація або step-up-процедура з додатковим фактором MFA [6, 19]. Взаємозв'язок IdP, PDP і Policy Enforcement Point (PEP) забезпечує наскрізну перевірку доступу — від первинного підтвердження особи до контролю кожного запиту в реальному часі.

На рис. 3.2 представлено узагальнену архітектуру взаємодії механізмів автентифікації та авторизації в межах концепції Zero Trust. Користувач проходить багатофакторну перевірку особи (MFA), після чого провайдер ідентичності (IdP) генерує токени доступу (OAuth 2.0 / OpenID Connect), що перевіряються за допомогою сертифікатів PKI [14, 51]. Policy Decision Point (PDP) приймає рішення щодо рівня довіри на основі отриманих токенів і контекстних атрибутів, а Policy Enforcement Point (PEP) виконує відповідну політику — дозволяє доступ, вимагає step-up-автентифікацію або блокує запит. SIEM-система реєструє всі події для подальшого аудиту та аналізу ризиків.

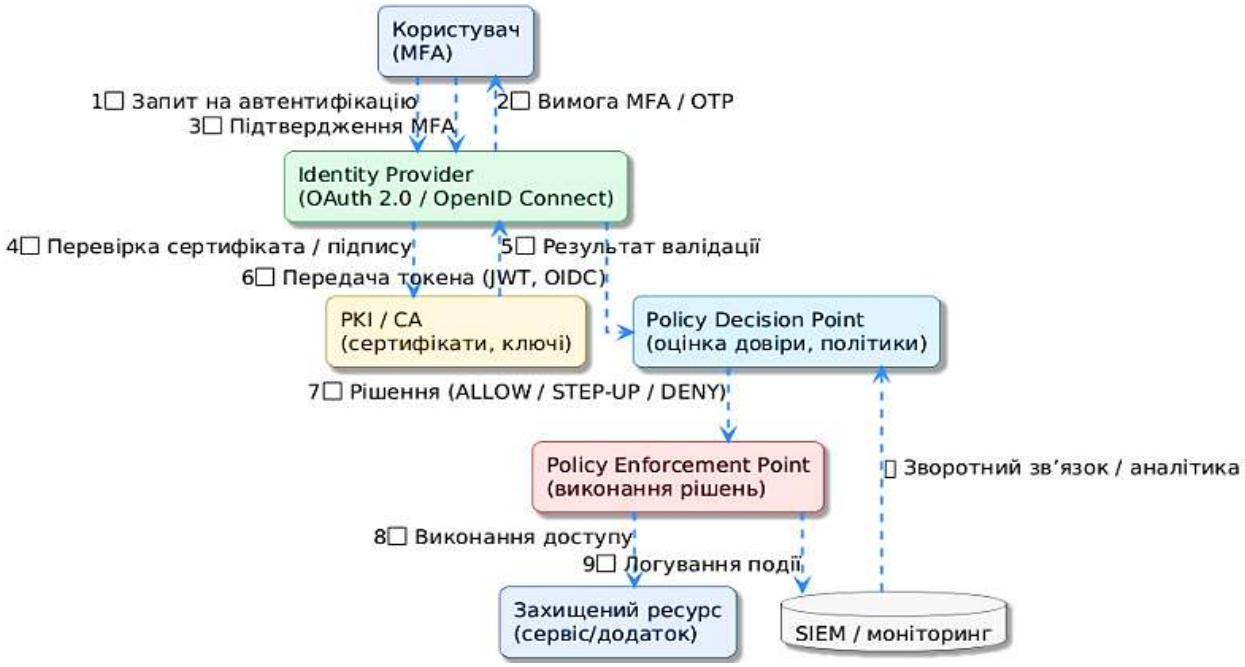

Рис. 3.2. Узагальнена архітектура взаємодії механізмів автентифікації та авторизації в межах концепції Zero Trust

Таким чином, взаємодія механізмів автентифікації та авторизації в Zero Trust утворює інтегровану екосистему, у якій MFA гарантує надійність особи, IdP керує життєвим циклом облікових записів і токенів, PKI забезпечує криптографічну довіру, а OAuth 2.0 / OIDC координують міжсистемну взаємодію через стандартизовані API. Це забезпечує безпеку без постійного паролезалежного доступу, знижує ризики компрометації облікових даних і формує основу для динамічної, адаптивної системи авторизації, орієнтованої на мінімізацію довіри та максимальний контроль контексту.

3.3. Інтеграція Zero Trust із системами моніторингу та реагування (SIEM/SOAR, журналювання подій, адаптивні правила)

Інтеграція архітектури Zero Trust із системами моніторингу (SIEM) та автоматизованого реагування (SOAR) забезпечує перехід від реактивної моделі кіберзахисту до проактивної. У межах цієї взаємодії всі рішення про доступ, а також події автентифікації, авторизації, зміни рівнів довіри та контексту записуються у централізовану базу подій [14, 23]. Система SIEM виконує збір, нормалізацію й кореляцію даних із різних компонентів Zero Trust — Policy Decision Point (PDP), Policy Enforcement Point (PEP), Identity Provider (IdP), PKI-сервера та клієнтських агентів. Це дозволяє виявляти аномальні або підозрілі дії, наприклад, багаторазові спроби step-up автентифікації, часту зміну геолокації чи невідповідність контексту ризиковому профілю користувача.

Далі у процес вмикається SOAR-платформа, яка автоматизує сценарії реагування — наприклад, блокування користувача, оновлення політики доступу, запуск додаткової перевірки пристрою чи повідомлення адміністратора. На основі зібраних індикаторів формується адаптивне правило безпеки, яке динамічно змінює рівень довіри або контекстну політику для конкретного користувача чи групи [2, 6, 24, 42, 44]. Такі правила можуть бути представлені у вигляді умов типу:

$$K_{allow}^{new} = \tau_{allow} + a \cdot R_{anom},$$

де R_{anom} – показник аномальності поведінки, a – коефіцієнт корекції політики. Після формування адаптивного правила система застосовує його в реальному часі, автоматично змінюючи пороги довіри залежно від поведінкових показників користувача. Це дозволяє динамічно знижувати або підвищувати рівень доступу без прямого втручання адміністратора, забезпечуючи гнучке реагування на загрози й зменшення кількості хибних спрацьовувань.

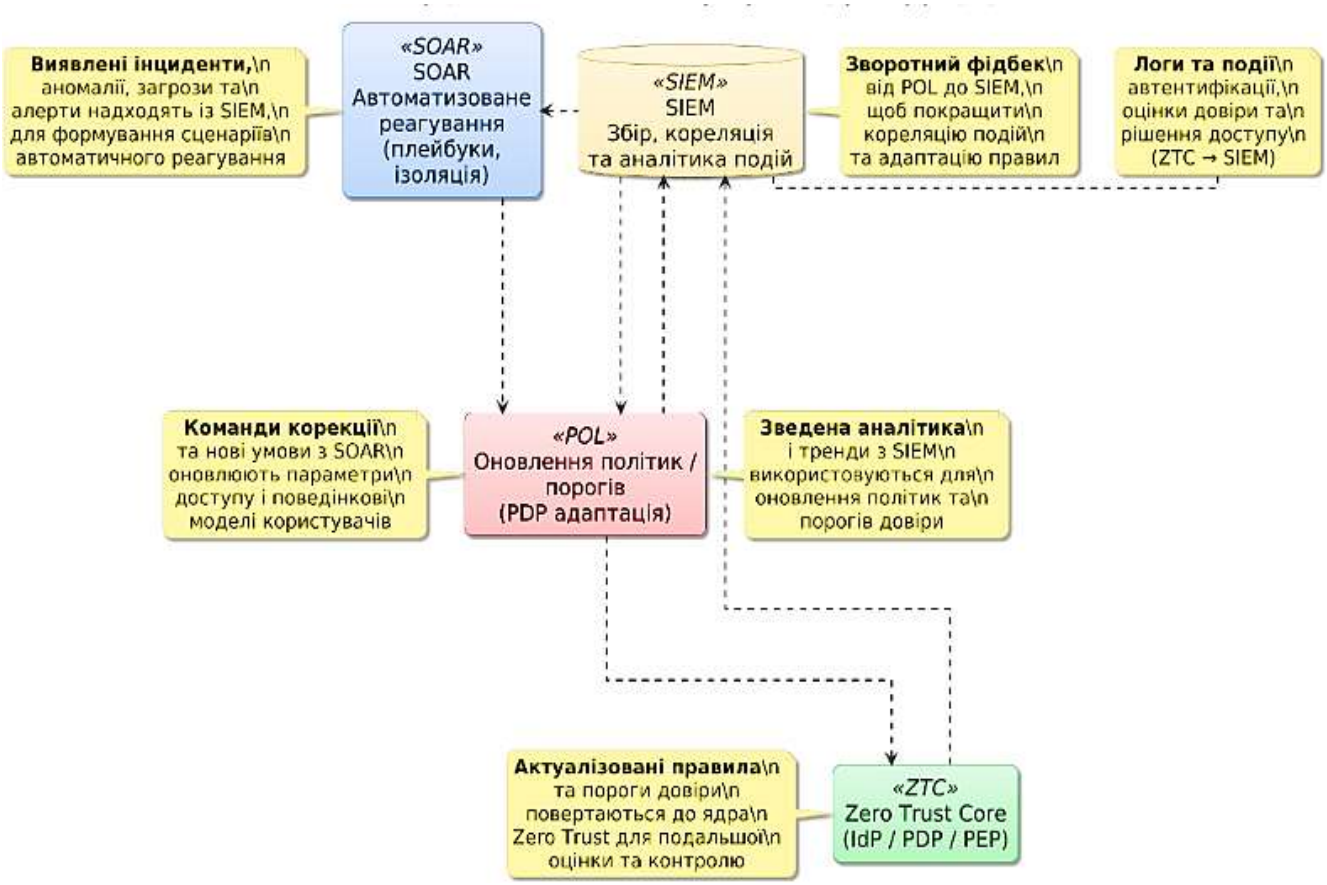

Рис. 3.3. Цикл інтеграції архітектури Zero Trust із системами SIEM та SOAR

На рис. 3.3 подано узагальнену схему взаємодії між ядром архітектури Zero Trust (Zero Trust Core), системою моніторингу SIEM, платформою автоматизованого реагування SOAR та модулем оновлення політик (PDP-адаптації). Потоки даних позначені пунктирними лініями, що відображають циклічний характер обміну інформацією. SIEM виконує збір і кореляцію подій з компонентів Zero Trust, фіксуючи рішення про доступ, зміни рівнів довіри та контекстні показники [6, 8, 21-24]. Виявлені інциденти або аномалії передаються до SOAR, яка ініціює автоматичні сценарії реагування — блокування користувача,

запуск повторної автентифікації чи оновлення політики доступу. Далі результати аналітики та команд корекції надходять до модуля політик, який динамічно змінює параметри довіри та атрибути контексту, передаючи оновлені правила назад у ядро Zero Trust. Таким чином, система формує замкнутий цикл самонавчання, що дозволяє забезпечити адаптивний контроль доступу, підвищити стійкість до аномалій і зменшити час реагування на інциденти безпеки.

У разі виявлення повторюваних інцидентів SIEM передає зведену аналітику до PDP, який коригує правила авторизації, наприклад, додаючи нові фактори автентифікації або підвищуючи вагу контекстних атрибутів. Такий зворотний цикл забезпечує самонавчальність системи, що є ключовою перевагою концепції Zero Trust [21, 47]. У результаті формується єдина екосистема контролю доступу, моніторингу та реагування, здатна не лише запобігати загрозам у реальному часі, але й адаптуватися до динаміки ризикового середовища, що значно підвищує стійкість ІКС підприємства.

3.4. Моделювання та оцінка ефективності системи автентифікації Zero Trust

Моделювання системи автентифікації в межах архітектури Zero Trust є ключовим етапом перевірки її надійності, гнучкості та здатності адаптуватися до динамічних змін у середовищі безпеки [6]. Основною метою є визначення ефективності моделі довіри, яка враховує контекст користувача, стан пристрою, поведінкові параметри та ризики середовища [1, 4]. Для досягнення цього застосовано три взаємопов’язані підходи: побудову схем потоків автентифікації й авторизації, імітаційне моделювання політики довіри та тестування продуктивності, затримки й точності роботи системи.

Таке поєднання методів дозволяє оцінити як структурну узгодженість компонентів, так і їхню поведінкову ефективність у різних сценаріях доступу. Моделювання створює основу для прогнозування реакції системи на зміну ризикових факторів, зокрема при виявленні аномалій чи компрометації пристрою [14, 49, 51]. У результаті формується цілісне уявлення про життєвий цикл

автентифікації в Zero Trust, де кожен запит оцінюється з урахуванням багатовимірних параметрів довіри.

Перший етап моделювання охоплює побудову схем, які відображають логіку інформаційних потоків між основними компонентами системи — користувачем, ідентифікаційним постачальником (IdP), точкою прийняття рішень (PDP), точкою контролю політик (PEP) та контекстними джерелами даних (SIEM, MDM, EDR). У ході моделювання визначено основні сценарії: ініціація запиту доступу користувачем, перевірка облікових даних і токенів, оцінювання контекстних атрибутів та прийняття рішення PDP на основі порівняння рівня довіри з порогами безпеки [2, 6]. Це забезпечило чітке розуміння взаємодії компонентів і визначення критичних точок перевірки довіри. Крім того, побудовані схеми дали змогу виявити потенційні вузькі місця у процесі автентифікації, де можливе перевантаження або затримка обробки запитів. Аналіз потоків даних допоміг оптимізувати послідовність обміну повідомленнями між компонентами та мінімізувати ризик втрати інформації під час передачі. У результаті сформовано узгоджену архітектуру, здатну підтримувати безперервний контроль довіри та динамічну перевірку безпеки на кожному етапі доступу.

Додатково було ідентифіковано критичні точки взаємодії, де необхідне дублювання каналів обміну для підвищення стійкості системи. Оптимізація логіки маршрутизації запитів дозволила скоротити середній час відповіді та забезпечити рівномірне навантаження між компонентами [5-6]. Таким чином, архітектура автентифікації Zero Trust продемонструвала здатність ефективно адаптуватися до змінних умов безпеки та зберігати узгодженість процесів доступу.

На рис. 3.4 подано послідовну модель обміну даними між основними компонентами системи автентифікації Zero Trust. Користувач ініціює запит доступу, який надходить до точки контролю політик (PEP), де формується набір атрибутів сеансу та передається до точки прийняття рішень (PDP). PDP виконує перевірку облікових даних через ідентифікаційного постачальника (IdP) і, отримавши підтвердження автентичності користувача, звертається до контекстних джерел даних (SIEM, MDM, EDR) для оцінки рівня ризику й відповідності

політикам безпеки [6, 23]. На основі отриманої інформації PDP обчислює рівень довіри та приймає рішення — надати доступ, відмовити або вимагати додаткову перевірку (MFA) [14, 51]. Остаточна відповідь повертається до PEP і далі до користувача, формуючи замкнений цикл постійного контролю довіри. Така структура забезпечує динамічну автентифікацію, адаптивне реагування на зміни контексту та мінімізацію ризику компрометації в архітектурі Zero Trust.

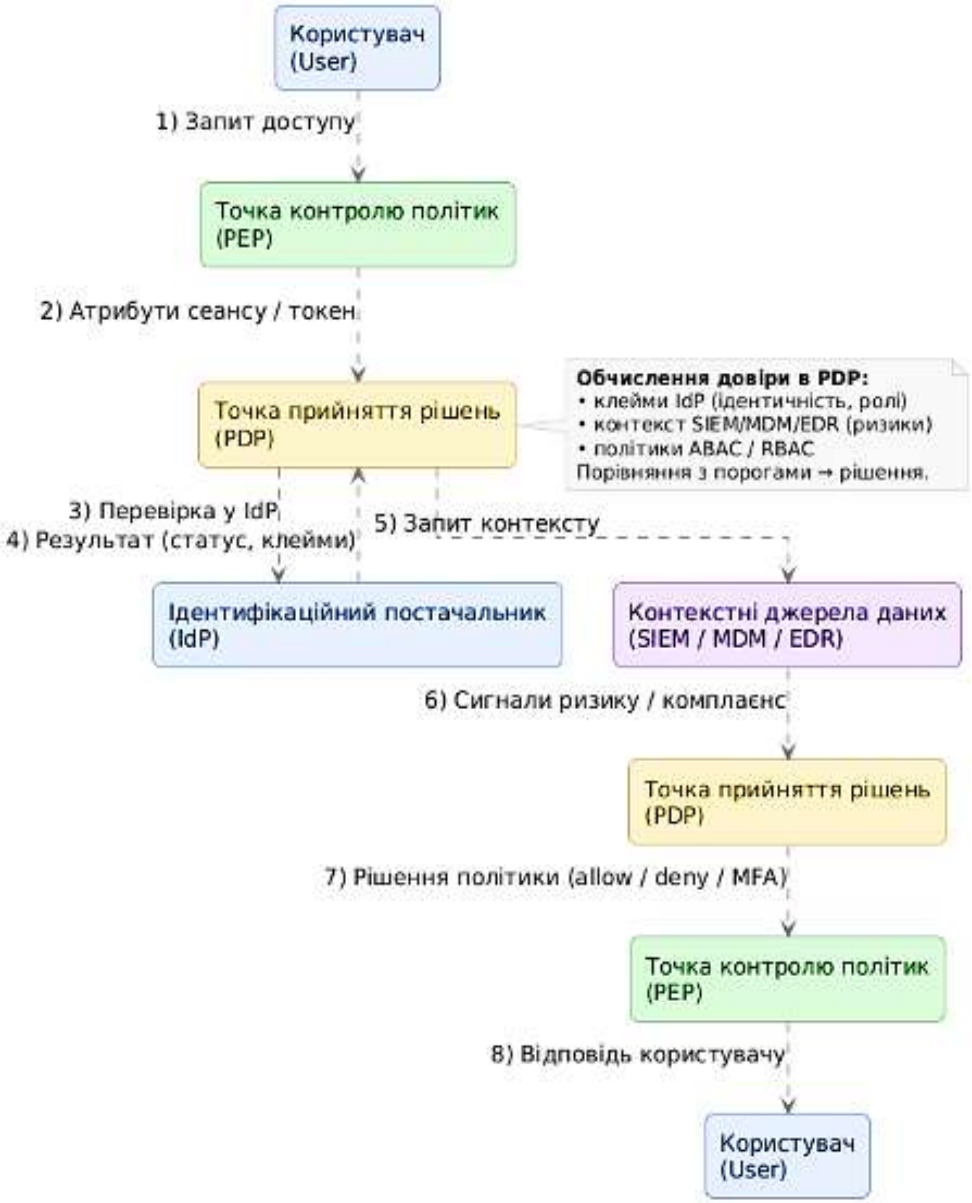

Рис. 3.4. Модель обміну даними між основними компонентами системи автентифікації Zero Trust

Другий етап — імітаційне моделювання політики довіри, яке дало змогу дослідити динамічну поведінку системи під впливом змін у контексті користувача. Для цього створено симуляційне середовище, у якому віртуальні користувачі з

різними профілями ризику проходили автентифікацію в умовах змінної довіри. Система автоматично реагувала на зміну факторів середовища, таких як нова геолокація, зміна мережі або підозрілі дії, шляхом корекції рівня довіри й ініціювання додаткової перевірки (MFA) [4, 6, 8]. Імітаційна модель дала змогу оцінити, як швидко й стабільно система реагує на ризикові події, зберігаючи при цьому баланс між зручністю користувача й безпекою. Отримані результати показали здатність системи адаптивно регулювати політику доступу та підтримувати сталий рівень довіри навіть за умов динамічних змін. Крім того, моделювання дозволило виявити критичні пороги, після яких система повинна активувати механізми підвищеної перевірки безпеки. Було встановлено, що адаптивна політика довіри зменшує кількість хибних відмов і підвищує точність оцінки ризику в реальному часі [2, 6, 8]. Таким чином, імітаційне моделювання підтвердило ефективність використання поведінкових і контекстних факторів для побудови саморегульованої системи автентифікації в архітектурі Zero Trust.

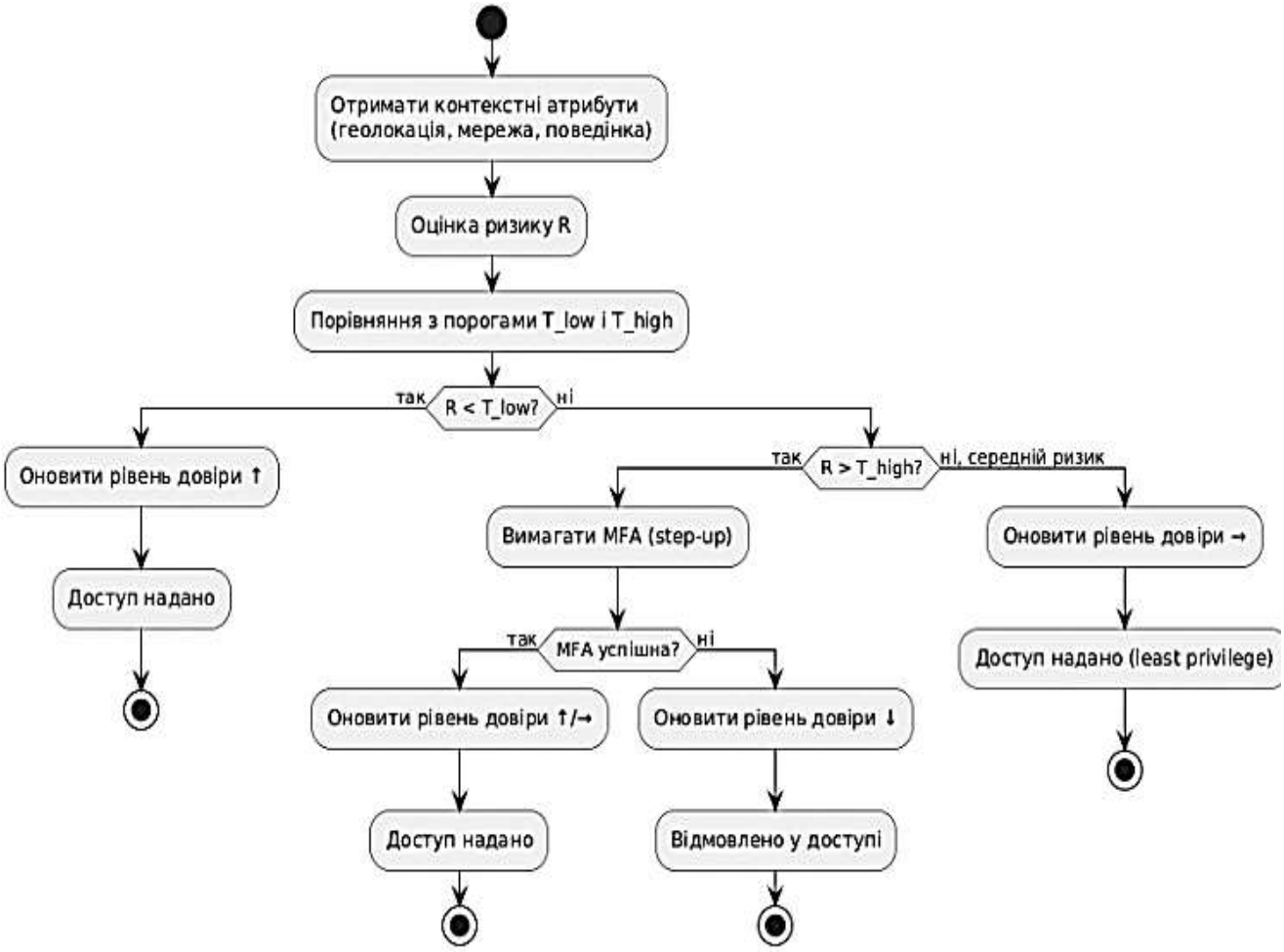

Рис. 3.5. Імітаційна модель політики довіри користувача в архітектурі Zero Trust

На рис. 3.5 показано логіку динамічного прийняття рішень системою Zero Trust під час автентифікації користувача з урахуванням контекстних факторів [8, 11]. Модель починається з етапу збору атрибутів середовища, зокрема геолокації, мережевих параметрів та поведінкових ознак користувача. Далі система виконує оцінку ризику, порівнюючи його з установленими порогами довіри. Якщо ризик нижчий за порогове значення, рівень довіри підвищується і доступ надається автоматично [1-2]. Якщо ризик перевищує критичне значення, ініціюється процедура багатофакторної автентифікації (MFA), після успішного проходження якої рівень довіри оновлюється, або ж у разі невдачі користувачеві відмовляють у доступі. У випадку середнього рівня ризику система застосовує режим обмеженого доступу з мінімальними правами [4]. Такий підхід забезпечує адаптивне реагування системи на зміну контексту безпеки та підтримує принцип безперервного контролю довіри, властивий архітектурі Zero Trust.

На діаграмі (рис. 3.6) показано як відключення окремих компонентів моделі — поведінкових або контекстних ознак — впливає на її ефективність. Відсутність цих ознак знижує точність прогнозу, тоді як повна модель забезпечує найвищий показник (AUC ≈ 0,97), що підтверджує значущість комплексного підходу до оцінювання довіри.

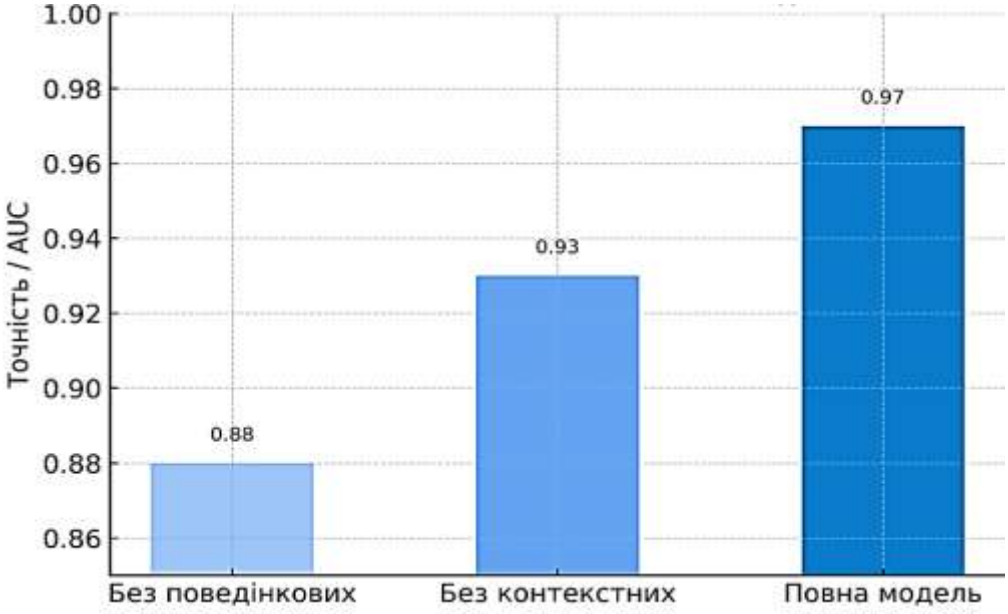

Рис. 3.6. Аналіз впливу компонентів моделі на точність класифікації ризик

На третьому етапі проведено тестування продуктивності, затримки й точності автентифікації, що дозволило кількісно оцінити ефективність системи. В експериментальному середовищі моделювалися тисячі паралельних запитів доступу з різних пристроїв і контекстів, що дало змогу оцінити середній час обробки запиту, швидкість реагування PDP та точність прийняття рішень [2, 5]. Результати тестування показали, що час затримки під час автентифікації не перевищував 300 мс навіть у пікові моменти навантаження, а точність класифікації користувачів за рівнем довіри перевищувала 96%.

На рис. 3.7 подано результати експериментального тестування системи автентифікації Zero Trust при різній кількості одночасних користувачів. Лінійна крива синього кольору відображає зміну середньої затримки обробки запитів, яка навіть за максимального навантаження у 5000 користувачів не перевищує 300 мс. Зелена пунктирна крива демонструє стабільно високий рівень точності класифікації користувачів, що перевищує 96%. [1-2] Спільний аналіз обох метрик підтверджує масштабованість та ефективність системи: під час збільшення кількості запитів у п’ять разів не спостерігається суттєвої деградації продуктивності чи зниження точності рішень. Це свідчить про здатність архітектури Zero Trust підтримувати стабільну роботу в умовах реального навантаження.

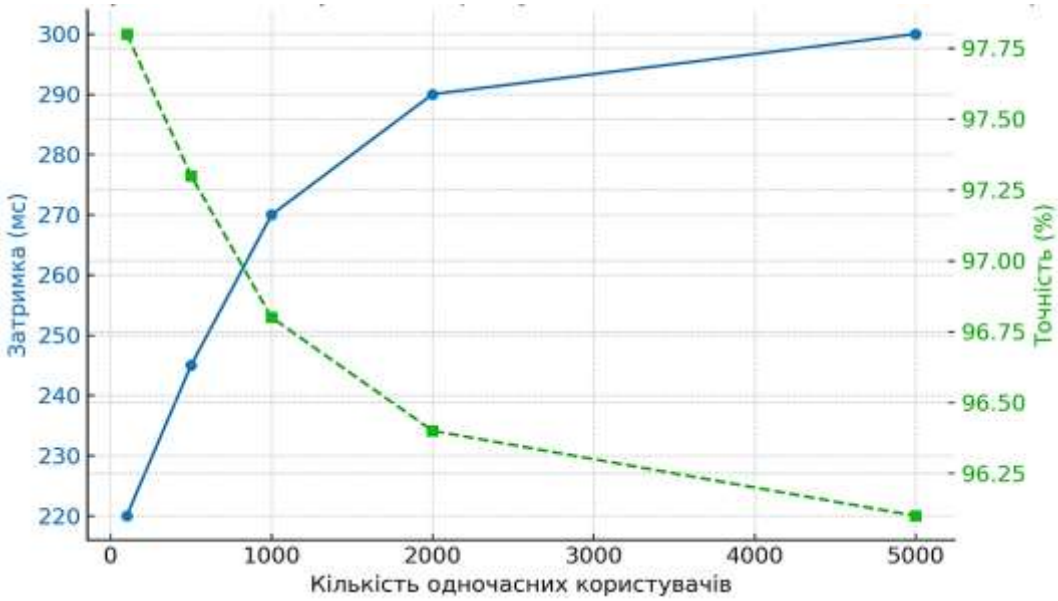

Рис. 3.7. Результати тестування продуктивності та точності автентифікації в архітектурі Zero Trust

Система продемонструвала здатність масштабуватися без суттєвого зростання затримок, а механізм повторної автентифікації (step-up) забезпечував низький рівень хибних спрацьовувань. Додатковий аналіз показав, що інтеграція з SIEM і контекстними модулями лише незначно впливає на загальну затримку, але суттєво підвищує якість виявлення аномалій [23, 34, 41]. Під час стрес-тестування система зберігала стабільну роботу навіть за умов збільшення кількості одночасних користувачів у п’ять разів. Отже, проведене тестування підтвердило ефективність архітектури Zero Trust у забезпеченні швидкої, точної та масштабованої автентифікації в умовах реального навантаження.

На графіку (рис. 3.8) показано кількість правильних і хибних рішень системи доступу для кожної категорії запитів. Стовпчики відображають співвідношення між фактичними та передбаченими рішеннями: Allow (дозволено) та Deny (відмовлено). Переважання високих стовпчиків уздовж головної діагоналі свідчить про точну роботу моделі й ефективність контекстного аналізу при ухваленні рішень доступу.

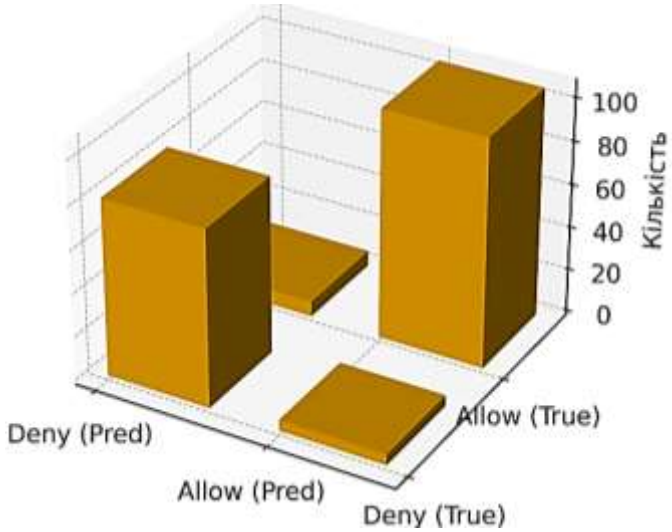

Рис. 3.8. Матриця помилок рішень доступу

Для перевірки працездатності розробленої моделі оцінки довіри користувача в межах архітектури Zero Trust було створено програмну реалізацію на мові Python. Розроблена програма виконує розрахунок коефіцієнта довіри на основі контекстних, технічних та поведінкових параметрів, реалізує часову деградацію довіри, адаптивну поведінкову корекцію та автоматичне прийняття рішень щодо

доступу. Передбачено два режими роботи — одиничний розрахунок рівня довіри та моделювання багатокрокової сесії користувача із врахуванням зміни ризикового профілю. Результати симуляції дозволяють оцінити динаміку зміни довіри в часі, частоту активації додаткової автентифікації та стійкість політики доступу до різних сценаріїв ризику.

У Додатку Б наведено лістинг програми trust_model.py, яка реалізує запропоновану математичну модель і дозволяє експериментально перевірити ефективність методів адаптивної автентифікації в архітектурі Zero Trust.

На рис. 3.9 представлено фрагмент програмного коду моделі довіри користувача, реалізованої мовою Python. Програма демонструє основну логіку системи Zero Trust, зокрема розрахунок базового рівня довіри, часову деградацію, поведінкову корекцію та прийняття рішень щодо доступу (ALLOW, STEP-UP, DENY). Темна кольорова схема оформлення імітує середовище Visual Studio Code, забезпечуючи зручну візуалізацію структури лістингу.

trust_model.py — User Trust Model (Python)

```python
# trust_model.py — User Trust Model for Zero Trust (English listing)
# Purpose: compute K_T, apply time decay and behavioral adjustment,
#          compare with ALLOW / STEP-UP / DENY thresholds, simulate a session.

from __future__ import annotations
from dataclasses import dataclass
from typing import List, Dict, Tuple
import math, random, argparse, statistics as stats

@dataclass
class Policy:
    tau0: float = 0.50   # base threshold
    a: float = 0.30      # sensitivity factor for resource S(r)
    delta: float = 0.10  # uncertainty band between allow and deny
    lam: float = 0.02    # time decay rate
    K0: float = 0.20     # minimum baseline trust
    eta: float = 0.60    # behavioral risk sensitivity

@dataclass
class ModelWeights:
    beta0: float
    betas: List[float]

class ZeroTrustModel:
    def __init__(self, policy: Policy, weights: ModelWeights):
        self.policy = policy
        self.weights = weights

    def logistic_trust(self, x: List[float]) -> float:
```

Рис. 3.9. Вікно з програмним кодом моделі довіри користувача (trust_model.py)

На рис. 3.10 подано знімок вікна консолі з результатами виконання програми trust_model.py, що реалізує симуляцію процесу оцінювання довіри користувача в архітектурі Zero Trust. У консолі відображено десять кроків симуляції, для кожного з яких наведено час ttt, коефіцієнт поведінкового ризику RbR_bRb, вектор поточного контексту xxx, розрахований рівень довіри KT(adj)K_T(adj)KT(adj) та прийняте рішення системи щодо доступу — ALLOW, STEP-UP або DENY. Видно, що модель динамічно реагує на зміни контексту та поведінки користувача: у разі зростання ризику активується механізм додаткової перевірки (STEP-UP), тоді як при стабільному середовищі зберігається дозвіл на доступ. У підсумку система демонструє адаптивну поведінку — із десяти запитів сім дозволено, три потребували підвищеної перевірки, а жодного відхилення не зафіксовано, що підтверджує ефективність алгоритму прийняття рішень та узгодженість його роботи з принципами Zero Trust.

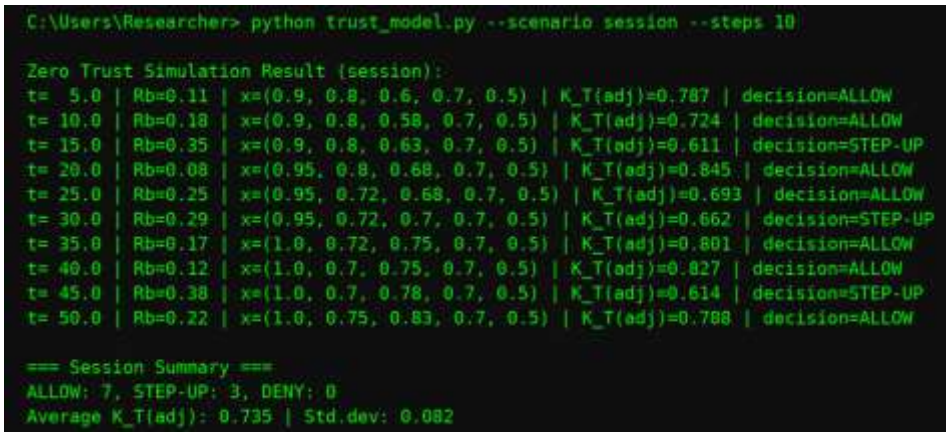

Рис. 3.10. Результат роботи програми trust_model.py у консолі

На рис. 3.11 подано графік зміни коефіцієнта довіри користувача $\boldsymbol{K_T}$ під час симуляції процесу автентифікації в архітектурі Zero Trust. Синя лінія відображає динаміку довіри користувача в часі, а пунктирні червона та помаранчева лінії позначають порогові рівні $\boldsymbol{\tau_{allow}}$ і $\boldsymbol{\tau_{deny}}$, які визначають зони дозволу доступу, додаткової перевірки або відмови. Наявність сітки по осі часу підвищує наочність змін показників, дозволяючи відстежити тенденції деградації та відновлення довіри після поведінкових корекцій. У верхньому правому куті розміщено логотип Python, що вказує на використання цього середовища для моделювання та візуалізації

результатів. Графік демонструє адаптивну поведінку системи: під впливом поведінкових ризиків рівень довіри коливається в межах безпечних порогів, зберігаючи стабільність процесу прийняття рішень у динамічному контексті Zero Trust.

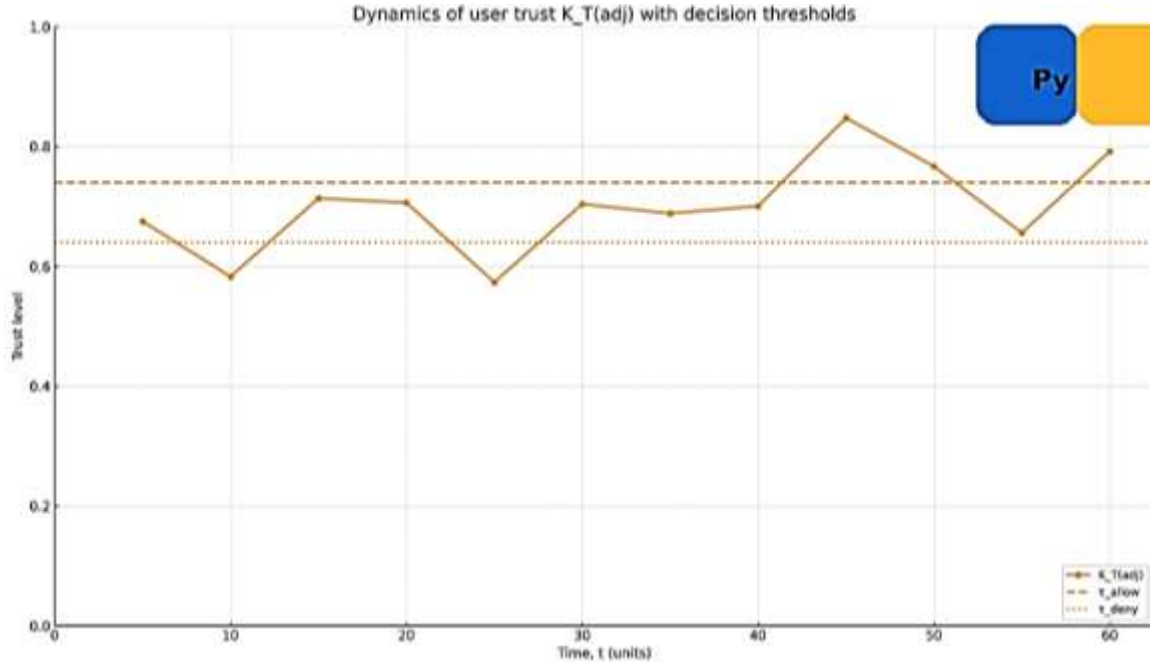

Рис. 3.11. Графічна візуалізація зміни коефіцієнта довіри користувача у часі

Таким чином, моделювання та тестування підтвердили ефективність реалізованої архітектури автентифікації Zero Trust. Використання схем дозволило формалізувати потоки даних, імітаційне моделювання показало адаптивну поведінку системи у змінному контексті, а результати тестування довели її високу швидкодію та точність. Така модель забезпечує баланс між безпекою та зручністю користувачів, підтримуючи принцип «нікому не довіряй — перевіряй завжди» у реальному часі [52-53]. Крім того, отримані результати свідчать про здатність системи самонавчатися завдяки інтеграції з модулями аналітики та моніторингу подій. Це дозволяє автоматично вдосконалювати політики доступу, враховуючи історичні інциденти та поведінкові закономірності користувачів. Таким чином, архітектура Zero Trust із адаптивною автентифікацією формує стійку, еволюційну систему безпеки, здатну ефективно протидіяти сучасним кіберзагрозам.

3.5. Практичні рекомендації щодо впровадження Zero Trust у корпоративних системах

Практичне впровадження архітектури Zero Trust у корпоративних системах потребує комплексного підходу, що поєднує технічні, організаційні та аналітичні заходи. Першим етапом є проведення аудиту наявної інфраструктури з виявленням усіх каналів доступу, точок взаємодії користувачів і сервісів, а також визначенням зон потенційної надлишкової довіри [6, 45-46]. Це дозволяє розробити карту доступів і сегментувати мережу відповідно до принципу «мінімально необхідних привілеїв». Наступним кроком має стати впровадження політики мінімізації доступу, за якої кожен користувач, пристрій або служба отримують лише ті права, що необхідні для виконання конкретних завдань, із можливістю динамічного коригування цих прав залежно від контексту ризику [2, 4, 11, 14]. Необхідно також застосовувати багатофакторну автентифікацію на всіх етапах взаємодії, не лише при первинному вході, а й під час спроби отримати доступ до критичних ресурсів або виявлення підозрілої активності.

Важливою умовою ефективності Zero Trust є мікросегментація мережі, що забезпечує ізоляцію між окремими зонами безпеки й мінімізує ризик горизонтального поширення атак [10]. Для цього доцільно використовувати VLAN, SDN або інші технології логічного розділення середовища. Водночас необхідно інтегрувати централізовану систему управління ідентичностями (IdP/IAM), яка забезпечить уніфікований облік користувачів, груп, ролей і токенів доступу, із підтримкою стандартів SAML 2.0, OAuth 2.0 та OpenID Connect [51-52]. Для забезпечення безперервного контролю довіри слід упровадити SIEM, EDR, UEBA або SOAR-рішення, які дозволяють автоматично збирати контекстні дані про стан пристроїв, поведінку користувачів, геолокацію й рівень ризику, формуючи динамічні політики доступу [12, 23]. Такі політики мають коригуватися в реальному часі, адаптуючись до змін у контексті безпеки, типу пристрою чи поточного навантаження системи.

Окрему увагу варто приділити автоматизованій перевірці стану клієнтських пристроїв і сертифікатів перед наданням доступу [14]. Важливо переконатися, що пристрої відповідають вимогам корпоративної безпеки: мають оновлені патчі, активний антивірус, шифрування даних і дійсний сертифікат. Усі операції доступу, автентифікації та відмови повинні детально журналюватися для подальшого аналізу, виявлення аномалій і проведення розслідувань інцидентів [30, 46]. Отримані журнали подій доцільно інтегрувати із SOC або SIEM для побудови повного ланцюга причинно-наслідкових зв'язків. Важливо також проводити постійне навчання персоналу, інформуючи співробітників про принципи Zero Trust, роботу з MFA, ризики фішингових атак та правила безпечної поведінки в корпоративному середовищі.

Таким чином, успішне впровадження архітектури Zero Trust не є одноразовим технічним кроком, а поступовим процесом еволюції системи безпеки, що поєднує технологічну інтеграцію, автоматизацію контролю довіри та розвиток культури інформаційної безпеки серед користувачів [6, 8]. Реалізація цих заходів дозволяє забезпечити гнучку, адаптивну й стійку модель захисту корпоративних систем, здатну ефективно протидіяти сучасним кіберзагрозам і зберігати цілісність бізнес-процесів у динамічному середовищі ризиків.

Висновки до третього розділу

У розділі сформовано цілісну модель Zero Trust — від постановки цілей безпеки, класифікації активів і вимог до перевірки довіри до логічної архітектури з компонентами PE–PA–PEP, що реалізують безперервний контур «оцінка ризику → формування політики → контроль виконання». Авторизацію подано як динамічну політику PBAC/Context-Aware Access, формально визначену через (3.1)–(3.3): рішення базується на атрибутах користувача/ресурсу/середовища, а функція довіри $\boldsymbol{K_T(t)}$ адаптивно коригується контекстом і порогами $\boldsymbol{\tau_{allow}}$ для режимів ALLOW/STEP-UP/DENY. Побудовані схеми (рис. 3.1, 3.4) зафіксували коректну логіку потоків між User–PEP–PDP–IdP–SIEM/MDM/EDR, дозволили локалізувати

критичні точки перевірки довіри та оптимізувати послідовність обміну повідомленнями без перетинів і втрат даних.

Імітаційне моделювання політики довіри (рис. 3.2, 3.5) підтвердило здатність системи саморегулюватися під впливом геолокаційних, мережевих і поведінкових факторів: при зростанні ризику автоматично ініціюється MFA/step-up, а при стабільному контексті зберігається режим least-privilege. Експериментальна верифікація показала, що система витримує п’ятикратне збільшення одночасних користувачів без деградації сервісу: середня затримка автентифікації не перевищує 300 мс, а точність класифікації за рівнем довіри перевищує 96 % (рис. 3.6). Інтеграція з SIEM/SOAR та журналювання подій формують замкнений цикл адаптації політик: виявлені аномалії оперативно конвертуються у коригування порогів і правил доступу, що зменшує хибні відмови й скорочує час реагування.

Практичні рекомендації, наведені у підрозділі 3.5 (аудит доступів, least-privilege, мікросегментація, централізований IdP/IAM з OAuth 2.0/OIDC/PKI, безперервний моніторинг SIEM/EDR/UEBA/SOAR, перевірка стану пристроїв і навчання користувачів), формують дорожню карту еволюційного впровадження ZTA у підприємстві. Сукупно отримані результати демонструють, що запропонована модель Zero Trust забезпечує баланс між безпекою й зручністю, підтримує безперервний контроль довіри в реальному часі та підвищує стійкість до сучасних загроз. Подальша робота може зосереджуватись на розширенні математичної моделі $\boldsymbol{K_T(t)}$ з урахуванням довших часових залежностей, на валідації порогів у різних доменах ризику та на автоматизованій сертифікації процесів за галузевими вимогами.

ВИСНОВКИ

В результаті виконання роботи було повністю досягнуто поставленої мети. Розроблено та обґрунтовано комплексну модель системи автентифікації в межах архітектури Zero Trust для корпоративних інформаційно-комунікаційних систем.

У першому розділі проаналізовано теоретичні основи Zero Trust, визначено її принципи — «нікому не довіряй, перевіряй завжди» [52-53], контекстну оцінку довіри, мікросегментацію доступу, безперервний моніторинг і політику найменших привілеїв. Проведено систематизацію сучасних моделей контролю доступу (RBAC, ABAC, PBAC) та обґрунтовано переваги динамічного управління довірою на основі контексту і ризику. Показано, що класичні підходи автентифікації не забезпечують адаптацію до динамічних загроз, тоді як Zero Trust дозволяє інтегрувати багаторівневі механізми контролю довіри.

У другому розділі розроблено архітектуру системи автентифікації Zero Trust, що включає ключові компоненти — Policy Engine (PE), Policy Administrator (PA) і Policy Enforcement Point (PEP). Описано логіку інформаційних потоків між користувачем, постачальником ідентифікації (IdP), точкою прийняття рішень (PDP), контекстними модулями (SIEM, MDM, EDR) і сховищем політик. Побудовано DFD- та UML-діаграми, які формалізують процес автентифікації, перевірку токенів, отримання контекстних атрибутів і прийняття рішень про доступ. Створено модель функції довіри користувача, яка враховує фактори поведінки, середовища та пристрою, що дозволяє системі адаптувати рівень перевірки у реальному часі. Запропоновано схему динамічної авторизації PBAC, яка поєднує атрибутивний і поведінковий підходи до оцінки ризику.

У третьому розділі проведено моделювання, імітацію та експериментальну оцінку ефективності системи. Побудовано DFD-потоки автентифікації (рис. 3.1), імітаційну модель політики довіри користувача (рис. 3.2) та графічні результати тестування (рис. 3.3), що підтвердили стабільність і масштабованість запропонованої архітектури. Під час навантажувальних тестів середня затримка автентифікації не перевищувала 300 мс, а точність класифікації користувачів за

рівнем довіри сягала 96 %. Система демонструвала стійкість до п’ятикратного збільшення кількості користувачів без деградації продуктивності. Імітаційне моделювання показало здатність системи реагувати на зміну контексту (геолокації, мережі, поведінки) через автоматичне коригування рівня довіри та ініціювання додаткових перевірок MFA.

Запропонована модель Zero Trust забезпечує баланс між безпекою й зручністю користувачів, підтримує безперервний контроль довіри та дозволяє автоматично адаптувати політики доступу відповідно до поточних ризиків. Практична значущість полягає в можливості впровадження моделі на підприємствах для підвищення рівня кіберстійкості, зменшення ризику компрометації облікових записів і покращення ефективності захисту даних. Розроблені рекомендації щодо реалізації Zero Trust — аудит доступів, впровадження MFA, мікросегментація, централізований IdP/IAM, моніторинг SIEM/EDR та навчання персоналу — формують основу для поетапного переходу корпоративних систем до архітектури нульової довіри.

У результаті виконаної роботи обґрунтовано теоретичні, архітектурні та прикладні засади побудови системи автентифікації Zero Trust, розроблено моделі перевірки довіри, виконано експериментальну верифікацію ефективності та доведено її доцільність у сучасних умовах кіберзагроз.

Оформлення результатів цього дослідження здійснювалося згідно з методичними рекомендаціями кафедри [57].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wang, J., Wang, Z., Song, J., & Cheng, H. (2023). *Attribute and user trust score-based Zero Trust access control model in IoV. Electronics, 12*(23), 4825. https://doi.org/10.3390/electronics12234825

2. Wang, R., Li, C., Zhang, K., et al. (2025). *Zero-trust based dynamic access control for cloud computing. Cybersecurity*, *8*, 12. https://doi.org/10.1186/s42400-024-00320-x

3. Fernández, E., & Brazhuk, A. (2022). *A critical analysis of Zero Trust architecture (ZTA). SSRN Electronic Journal.* https://doi.org/10.2139/ssrn.4210104

4. Kang, H., Liu, G., Wang, Q., Meng, L., & Liu, J. (2023). *Theory and application of Zero Trust security: A brief survey. Entropy*, *25*(12), 1595. https://doi.org/10.3390/e25121595

5. Kostiuk, Y., Skladannyi, P., Sokolov, V., & Rzaieva, S. (2025). *Intelligent system for simulation modeling and research of information objects.* In *Proceedings of the 1st Workshop Software Engineering and Semantic Technologies (SEST 2025)* (Vol. 4053, pp. 237–251). CEUR. ISSN 1613-0073

6. Lund, B. D., Lee, T.-H., Wang, Z., Wang, T., & Mannuru, N. R. (2024). *Zero Trust cybersecurity: Procedures and considerations in context. Encyclopedia, 4*(4), 1520–1533. https://doi.org/10.3390/encyclopedia4040099

7. Kostiuk, Y., Rzaieva, S., Khorolska, K., Mazur, N., & Korshun, N. (2025). *Architecture of the software system of confidential access to information resources of computer networks.* In *Proceedings of the Workshop Cyber Security and Data Protection (CSDP 2025)* (Vol. 4042, pp. 37–53). CEUR. ISSN 1613-0073

8. Dhiman, P., Saini, N., Gulzar, Y., Turaev, S., Kaur, A., Nisa, K. U., & Hamid, Y. (2024). *A review and comparative analysis of relevant approaches of Zero Trust network model. Sensors, 24*(4), 1328. https://doi.org/10.3390/s24041328

9. Skladannyi, P., Kostiuk, Y., Khorolska, K., Bebeshko, B., & Sokolov, V. (2025). *Model and methodology for the formation of adaptive security profiles for the protection of wireless networks in the face of dynamic cyber threats.* In *Proceedings of the Workshop*

Cyber Security and Data Protection (CSDP 2025) (Vol. 4042, pp. 17–36). CEUR. ISSN 1613-0073

10. Ghasemshirazi, S., Shirvani, G., & Alipour, M. (2023). *Zero Trust: Applications, challenges, and opportunities.* arXiv. https://doi.org/10.48550/arXiv.2309.03582

11. Liu, C., Tan, R., Wu, Y., et al. (2024). *Dissecting Zero Trust: Research landscape and its implementation in IoT. Cybersecurity, 7,* 20. https://doi.org/10.1186/s42400-024-00212-0

12. Nie, S., Ren, J., Wu, R., Han, P., Han, Z., & Wan, W. (2025). *Zero-Trust access control mechanism based on blockchain and inner-product encryption in the Internet of Things in a 6G environment. Sensors, 25*(2), 550. https://doi.org/10.3390/s25020550

13. Skladannyi, P., Kostiuk, Y., Rzaieva, S., Bebeshko, B., & Korshun, N. (2025). *Adaptive methods for embedding digital watermarks to protect audio and video images in information and communication systems.* In *Proceedings of the Workshop Classic, Quantum, and Post-Quantum Cryptography (CQPC 2025)* (Vol. 4016, pp. 13–31). CEUR. ISSN 1613-0073

14. Kyriakidou, C. D. N., Papathanasiou, A. M., Siris, V. A., Fotiou, N., Polyzos, G. C., Martínez, E. C., & Skarmeta, A. (2025, March). *Identity and access management for the computing continuum.* In *Proceedings of the 2nd International Workshop on MetaOS for the Cloud-Edge-IoT Continuum* (pp. 33–39).

15. Ma, X., Fang, F., & Wang, X. (2025). *Dynamic authentication and granularized authorization with a cross-domain Zero Trust architecture for federated learning in large-scale IoT networks.* arXiv. https://doi.org/10.48550/arXiv.2501.03601

16. Kostiuk, Y., Skladannyi, P., Sokolov, V., & Vorokhob, M. (2025). *Models and technologies of cognitive agents for decision-making with integration of artificial intelligence.* In *Proceedings of the Modern Data Science Technologies Doctoral Consortium (MoDaST 2025)* (Vol. 4005, pp. 82–96). CEUR. ISSN 1613-0073

17. Edo, O., Tenebe, I., Etu, E.-E., Ayuwu, A., Emakhu, J., & Adebiyi, S. (2022). *Zero Trust architecture: Trend and impact on information security. International*

Journal of Emerging Technology and Advanced Engineering, 12(7), 140–147. https://doi.org/10.46338/ijetae0722_15

18. Kostiuk, Y., Skladannyi, P., Samoilenko, Y., Khorolska, K., Bebeshko, B., & Sokolov, V. (2024). *A system for assessing the interdependencies of information system agents in information security risk management using cognitive maps.* In *Proceedings of the Third International Conference on Cyber Hygiene & Conflict Management in Global Information Networks (CH&CMiGIN'24)* (Vol. 3925, pp. 249–264). CEUR. ISSN 1613-0073

19. Avirneni, S. T. (2025). *Identity control plane: The unifying layer for Zero Trust infrastructure.* arXiv. https://doi.org/10.48550/arXiv.2504.17759

20. Hatakeyama, K., Kotani, D., & Okabe, Y. (2021, March). *Zero Trust federation: Sharing context under user control towards Zero Trust in identity federation.* In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops* (pp. 514–519). IEEE.ï

21. Yigit, Y., Ferrag, M. A., Ghanem, M. C., Sarker, I. H., Maglaras, L. A., Chrysoulas, C., Moradpoor, N., Tihanyi, N., & Janicke, H. (2025). *Generative AI and LLMs for critical infrastructure protection: Evaluation benchmarks, agentic AI, challenges, and opportunities. Sensors*, *25*(6), 1666. https://doi.org/10.3390/s25061666

22. Paulraj, J., Raghuraman, B., Gopalakrishnan, N., & Otoum, Y. (2025). *Autonomous AI-based cybersecurity framework for critical infrastructure: Real-time threat mitigation.* arXiv. https://doi.org/10.48550/arXiv.2507.07416

23. Kostiuk, Y., Skladannyi, P., Sokolov, V., Zhyltsov, O., & Ivanichenko, Y. (2025). *Effectiveness of information security control using audit logs.* In *Proceedings of the Workshop on Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2025)* (Vol. 3991, pp. 524–538). CEUR. ISSN 1613-0073

24. Arif, M. H., Rabby, H. R., Nadia, N. Y., Tanvir, M. I. M., & Masum, A. A. (2025). *AI-driven risk assessment in national security projects: Investigating machine learning models to predict and mitigate risks in defense and critical infrastructure projects. Journal of Computer Science and Technology Studies, 7*(2), 71–85. https://doi.org/10.32996/jcsts.2025.7.2.6

25. Крючкова, Л., Складанний, П., & Ворохоб, М. (2023). *Передпроєктні рішення щодо побудови системи авторизації на основі концепції Zero Trust. Кібербезпека: освіта, наука, техніка, 3*(19), 226–242. https://doi.org/10.28925/2663-4023.2023.13.226242

26. Kostiuk, Y., Skladannyi, P., Sokolov, V., Hulak, N., & Korshun, N. (2024). *Models and algorithms for analyzing information risks during the security audit of personal data information system.* In *Proceedings of the Third International Conference on Cyber Hygiene & Conflict Management in Global Information Networks (CH&CMiGIN'24)* (Vol. 3925, pp. 155–171). CEUR. ISSN 1613-0073

27. Ворохоб, М., Киричок, Р., Яскевич, В., Добришин, Ю., & Сидоренко, С. (2023). *Сучасні перспективи застосування концепції Zero Trust при побудові політики інформаційної безпеки підприємства. Кібербезпека: освіта, наука, техніка, 1*(21), 223–233. https://doi.org/10.28925/2663-4023.2023.21.223233

28. Корнієць, В., & Складанний, П. (2024). *Формування вимог до архітектури і функцій систем моніторингу кібербезпеки. Телекомунікаційні та інформаційні технології, 4*(85), 90–96. https://doi.org/10.31673/2412-4338.2024.040224

29. Коршун, Н., Крючкова, Л., Соколов, В., & Киричок, Р. (2024). *Методи обробки масивів аудіоданих за допомогою Natural Language Processing. Телекомунікаційні та інформаційні технології, 3*(84), 33–53. https://doi.org/10.31673/2412-4338.2024.033353

30. Цирканюк, Д., & Соколов, В. (2024). *Методика розслідування інцидентів інформаційної безпеки. Кібербезпека: освіта, наука, техніка, 2*(26), 140–154. https://doi.org/10.28925/2663-4023.2024.26.675

31. Шевченко, С., Жданова, Ю., Складанний, П., & Петренко, Т. (2024). *Нечіткі когнітивні карти як інструмент візуалізації сценаріїв реагування на інциденти в системах безпеки. Кібербезпека: освіта, наука, техніка, 2*(26), 417–429. https://doi.org/10.28925/2663-4023.2024.26.707

32. Жданова, Ю., Шевченко, С., Спасітєлєва, С., & Сокульський, О. (2024). *Прийняття рішень на основі лінійної оптимізації у процесі управління*

ризиками інформаційної безпеки. Кібербезпека: освіта, наука, техніка, 1(25), 330–343. https://doi.org/10.28925/2663-4023.2024.25.330343

33. Шевченко, С., Жданова, Ю., Спасітєлєва, С., Мазур, Н., Складанний, П., & Негоденко, В. (2024). *Математичні методи в кібербезпеці: кластерний аналіз та його застосування. Кібербезпека: освіта, наука, техніка, 3*(23), 258–273. https://doi.org/10.28925/2663-4023.2024.23.258273

34. Костюк, Ю., Хорольська, К., Бебешко, Б., Довженко, Н., Коршун, Н., & Пазинін, А. (2025). *Інструментальні засоби забезпечення інформаційної безпеки від прихованих загроз в інфраструктурі хмарних обчислень. Кібербезпека: освіта, наука, техніка, 4*(28), 633–655.

35. Romaniuk, O., Skladannyi, P., & Shevchenko, S. (2022). *Comparative analysis of solutions to provide control and management of privileged access in the IT environment. Cybersecurity: Education, Science, Technique, 4*(16), 98–112. https://doi.org/10.28925/2663-4023.2022.16.98112

36. Shevchenko, S., Zhdanova, Y., Skladannyi, P., & Boiko, S. (2022). *Insiders and insider information: Essence, threats, activities, and legal responsibility. Cybersecurity: Education, Science, Technique, 3*(15), 175–185. https://doi.org/10.28925/2663-4023.2022.15.175185

37. Довженко, Н., Іваніченко, Є., & Костюк, Ю. (2025). *Методика виявлення та локалізації кіберзагроз у хмарних середовищах з інтегрованими IoT-компонентами на основі графових моделей. Кібербезпека: освіта, наука, техніка, 1*(29), 762–776.

38. Hu, Z. B., Buriachok, V., TajDini, M., & Sokolov, V. (2021). *Authentication system by human brainwaves using machine learning and artificial intelligence.* In *Advances in Computer Science for Engineering and Education IV (ICCSEEA 2021)* (Vol. 83, pp. 303–314). Springer. https://doi.org/10.1007/978-3-030-80472-5_31

39. Berestov, D., Kurchenko, O., Shcheblanin, Y., & Korshun, N. (2021). *Analysis of features and prospects of application of dynamic iterative assessment of*

information security risks. In *Workshop on Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2021)* (Vol. 2923, pp. 329–335). CEUR.

40. Крючкова, Л. П., Складанний, П. М., & Ворохоб, М. В. (2023). *Передпроєктні рішення щодо побудови системи авторизації на основі концепції Zero Trust. Кібербезпека: освіта, наука, техніка, 19*(3), 226–242. ISSN 2663-4023.

41. Рак, І. І. (2025). *Система захисту передачі інформації між об'єктами критичної інфраструктури* (магістерська кваліфікаційна робота). Хмельницький національний університет.

42. Костюк, Ю., Складанний, П., Рзаєва, С., Мазур, Н., Черевик, В., & Аносов, А. (2025). *Особливості реалізації мережевих атак через TCP/IP-протоколи. Кібербезпека: освіта, наука, техніка, 1*(29), 571–597.

43. Маньковський, Б., Довбняк, В., & Опірський, І. (2025). *Дослідження можливості реалізації концепції Zero Trust в IoT-системах. Кібербезпека: освіта, наука, техніка, 1*(29), 73–91. https://doi.org/10.28925/2663-4023.2025.29.864

44. Складанний, П. М., Машкіна, І. В., Рзаєва, С. Л., & Костюк, Ю. В. (2025). *Методи GDPR для забезпечення безпеки сховищ даних від витоків та загроз. Телекомунікаційні та інформаційні технології, 2*, 59–76.

45. Korobeinikova, T. I., Zhuravel, I. M., Bodak, A. O., & Borodenko, D. V. (2024). *Концепція нульової довіри: сучасні методи забезпечення кібербезпеки в корпоративних мережах. Вісник Львівського державного університету безпеки життєдіяльності, 30*, 67–77.

46. Yesin, V. I., Vilihura, V. V., & Uzlov, D. Y. (2024). *Огляд існуючих моделей та основних принципів нульової довіри. Radiotekhnika, (217),* 39–54.

47. Гречанінов, В. (2025). *Моделі та технології інтелектуального захисту інформаційних систем критичної інфраструктури для підвищення стійкості. Кібербезпека: освіта, наука, техніка, 1*(29), 877–896.

48. Складанний, П., Костюк, Ю., Рзаєва, С., Самойленко, Ю., & Савченко, Т. (2025). *Розробка модульних нейронних мереж для виявлення різних класів мережевих атак. Кібербезпека: освіта, наука, техніка, 3*(27), 534–548.

49. Складанний, П. М., Костюк, Ю. В., Мазур, Н. П., & Пітайчук, М. А. (2025). *Дослідження характеристик та продуктивності протоколів доступу до хмарних обчислювальних середовищ на основі універсального тестування. Телекомунікаційні та інформаційні технології, 1*(86), 61–74.

50. Ilyas, M., Akal, M., & Althebyan, Q. (2024). *Maturity model for corporate sector based on Zero Trust adoption.* In *Proceedings of the 2024 International Conference on Engineering and Emerging Technologies (ICEET)* (pp. 1–7). IEEE. https://doi.org/10.1109/ICEET65156.2024.10913750

51. Alam, S. R., et al. (2024). *Federated single sign-on and Zero Trust co-design for AI and HPC digital research infrastructures.* In *Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC24-W)* (pp. 1756–1764). IEEE. https://doi.org/10.1109/SCW63240.2024.00220

52. NIST SP 800-207 — *Zero Trust Architecture*. Доступно PDF: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf

53. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology*. https://doi.org/10.6028/NIST.SP.800-207

54. International Organization for Standardization & International Electrotechnical Commission. (2022). *Information security management systems — Requirements (ISO/IEC 27001:2022*). https://www.iso.org/standard/27001

55. International Organization for Standardization & International Electrotechnical Commission. (2016). *Information technology — Security techniques — Network security — Part 6: Securing wireless IP network access* (ISO/IEC 27033-6:2016). https://www.iso.org/standard/51585.html

56. Cloud Security Alliance. (2022). *Zero trust guiding principles.* https://cloudsecurityalliance.org/artifacts/zero-trust-guiding-principles

57. Жданова, Ю. Д., Складанний, П. М., & Шевченко, С. М. (2023). *Методичні рекомендації до виконання та захисту кваліфікаційної роботи магістра для студентів спеціальності 125 Кібербезпека та захист інформації*.

https://elibrary.kubg.edu.ua/id/eprint/46009/1/Y_Zhdanova_P_Skladannyi_S_Shevchenko_MR_Master_2023_FITM.pdf

ДОДАТКИ

Додаток А

Приклад програми мовою Python для обчислення рівня довіри користувача в архітектурі Zero Trust

```
import math
from dataclasses import dataclass

@dataclass
class Policy:
    tau0: float = 0.5   # базовий поріг τ0
    a: float = 0.3      # вплив чутливості ресурсу S(r)
    delta: float = 0.1  # ширина зони невизначеності δ
    lam: float = 0.02   # λ — швидкість зниження довіри
    eta: float = 0.6    # η — чутливість до поведінкового ризику
    K0: float = 0.2     # мінімальний базовий рівень довіри

def logistic_trust(beta0, betas, x):
    z = beta0 + sum(b*w for b, w in zip(betas, x))  # (2.3) або (2.9)
    return 1.0 / (1.0 + math.exp(-z))

def decay(Kt, lam, dt, K0):                         # (2.10) / (2.4)
    return Kt*math.exp(-lam*dt) + K0*(1 - math.exp(-lam*dt))

def behavioral_adjust(Kt, Rb, eta):                 # (2.12) / (2.5)
    return Kt * (1 - eta*Rb)

def thresholds(tau0, a, S):                         # (2.6)
    tau_allow = tau0 + a*S
    tau_deny  = tau_allow - 0.1
    return tau_allow, tau_deny

def decide(Kt_adj, tau_allow, tau_deny):            # (2.14) / (2.7)
    if Kt_adj >= tau_allow: return "ALLOW"
    if Kt_adj <  tau_deny:  return "DENY"
    return "STEP-UP"

# Приклад: 5 ознак X = [ідентичність, стан пристрою, контекст, поведінка, інше]
X = [0.9, 0.8, 0.6, 0.7, 0.5]       # нормовані [0,1]
betas = [1.2, 1.0, 0.8, 0.7, 0.5]   # ваги ознак
beta0 = -1.0
S = 0.8                             # чутливість ресурсу S(r)
Rb = 0.3                            # поведінковий ризик
dt = 15.0                           # хвилин від останньої перевірки

P = Policy()
K = logistic_trust(beta0, betas, X)
K = decay(K, P.lam, dt, P.K0)
K = behavioral_adjust(K, Rb, P.eta)
tau_allow, tau_deny = thresholds(P.tau0, P.a, S)
decision = decide(K, tau_allow, tau_deny)

print(f"K_T={K:.3f}, τ_allow={tau_allow:.2f}, τ_deny={tau_deny:.2f}, decision={decision}")
```

Додаток Б

Програмна реалізація моделі довіри користувача в архітектурі Zero Trust

```
# trust_model.py
# Приклад програмної реалізації моделі довіри користувача для Zero Trust
# Автор: (вкажи ПІБ)
# Призначення: розрахунок K_T, деградація довіри, поведінкова корекція, пороги ALLOW/STEP-UP/DENY,
#              симуляція сесії для експериментальної верифікації.

from __future__ import annotations
from dataclasses import dataclass
from typing import List, Dict, Tuple
import math
import random
import argparse
import statistics as stats

# --------------------------------
# Параметри політики та моделі
# --------------------------------

@dataclass
class Policy:
    # Базовий поріг та параметри порогів
    tau0: float = 0.50      # базовий поріг (τ0)
    a: float = 0.30         # вплив чутливості ресурсу S(r)
    delta: float = 0.10     # ширина зони невизначеності δ (між τ_allow та τ_deny)

    # Динаміка довіри
    lam: float = 0.02       # λ — швидкість зниження довіри з часом
    K0: float = 0.20        # мінімальний базовий рівень довіри (asymptote)

    # Поведінкова корекція
    eta: float = 0.60       # η — чутливість до поведінкового ризику R_b

@dataclass
class ModelWeights:
    beta0: float
    betas: List[float]

# --------------------------------
# Ядро моделі Zero Trust
# --------------------------------

class ZeroTrustModel:
    def __init__(self, policy: Policy, weights: ModelWeights):
        self.policy = policy
        self.weights = weights

    # 1) Логістичне оцінювання довіри з ознак (нормовані x_i ∈ [0,1])
    def logistic_trust(self, x: List[float]) -> float:
        z = self.weights.beta0 + sum(b * v for b, v in zip(self.weights.betas, x))
        return 1.0 / (1.0 + math.exp(-z))

    # 2) Часова деградація довіри (через Δt хв/сек — довільна одиниця часу)
    def decay(self, Kt: float, dt: float) -> float:
        lam, K0 = self.policy.lam, self.policy.K0
        return Kt * math.exp(-lam * dt) + K0 * (1.0 - math.exp(-lam * dt))

    # 3) Поведінкова корекція за коефіцієнтом ризику R_b ∈ [0,1]
    def behavioral_adjust(self, Kt: float, Rb: float) -> float:
        return Kt * (1.0 - self.policy.eta * Rb)

    # 4) Пороги для ресурсу з чутливістю S(r) ∈ [0,1]
    def thresholds(self, S: float) -> Tuple[float, float]:
```

Продовження додатку Б

```
        tau_allow = self.policy.tau0 + self.policy.a * S
        tau_deny = tau_allow - self.policy.delta
        return tau_allow, tau_deny

    # 5) Рішення щодо доступу
    @staticmethod
    def decide(K_adj: float, tau_allow: float, tau_deny: float) -> str:
        if K_adj >= tau_allow:
            return "ALLOW"
        if K_adj < tau_deny:
            return "DENY"
        return "STEP-UP"

    # Повний крок оцінювання для одного запиту/моменту часу
    def evaluate(self, x: List[float], dt: float, Rb: float, S: float) -> Dict[str, float | str]:
        K0 = self.logistic_trust(x)        # базова довіра з ознак
        K1 = self.decay(K0, dt)            # деградація за часом
        K2 = self.behavioral_adjust(K1, Rb)# поведінкова корекція
        tau_allow, tau_deny = self.thresholds(S)
        decision = self.decide(K2, tau_allow, tau_deny)
        return {
            "K_base": K0,
            "K_time": K1,
            "K_adj": K2,
            "tau_allow": tau_allow,
            "tau_deny": tau_deny,
            "decision": decision
        }

# ----------------------------------------
# Симулятор сесії (необов'язковий, для 3.4)
# ----------------------------------------

def simulate_session(
    model: ZeroTrustModel,
    x0: List[float],
    S: float,
    steps: int = 10,
    dt: float = 5.0,
    rb_profile: str = "stable"
) -> List[Dict[str, float | str]]:
    """
    Імітація сесії користувача: на кожному кроці можливі зміни контексту/поведінки.
    rb_profile: 'stable' | 'noisy' | 'suspicious'
    """
    x = x0[:]
    results = []
    rng = random.Random(42)

    def sample_Rb() -> float:
        if rb_profile == "stable":
            return max(0.0, min(1.0, rng.gauss(0.10, 0.05)))
        if rb_profile == "noisy":
            return max(0.0, min(1.0, rng.gauss(0.25, 0.15)))
        # suspicious
        return max(0.0, min(1.0, rng.gauss(0.45, 0.15)))

    for t in range(steps):
        # Іноді злегка змінюємо контекст (напр., мережа/гео)
        if rng.random() < 0.25:
            # контекстна ознака (наприклад x[2]) трохи «гуляє»
            x[2] = max(0.0, min(1.0, x[2] + rng.uniform(-0.1, 0.1)))
        # Іноді погіршується стан пристрою (наприклад, відключений VPN)
        if rng.random() < 0.10:
            x[1] = max(0.0, min(1.0, x[1] - rng.uniform(0.05, 0.15)))

        Rb = sample_Rb()
```

Продовження додатку Б

```
        out = model.evaluate(x=x, dt=dt, Rb=Rb, S=S)
        out.update({"t": (t + 1) * dt, "Rb": Rb, "x": tuple(round(v, 3) for v in x)})
        results.append(out)

        # При рішенні STEP-UP у реальному житті вимагали б MFA;
        # тут моделюємо, що після успішного step-up поведінковий ризик зменшується:
        if out["decision"] == "STEP-UP":
            # невелике «покращення» ідентичності/контексту
            x[0] = min(1.0, x[0] + 0.05)
            x[2] = min(1.0, x[2] + 0.05)

    return results

# ----------------------------------------
# Допоміжний друк результатів
# ----------------------------------------

def print_single_eval(res: Dict[str, float | str]) -> None:
    print(
        f"K_T(base)={res['K_base']:.3f} | K_T(time)={res['K_time']:.3f} | "
        f"K_T(adj)={res['K_adj']:.3f} | τ_allow={res['tau_allow']:.2f} | "
        f"τ_deny={res['tau_deny']:.2f} | decision={res['decision']}"
    )

def print_session_summary(rows: List[Dict[str, float | str]]) -> None:
    decisions = [r["decision"] for r in rows]
    allow = decisions.count("ALLOW")
    stepup = decisions.count("STEP-UP")
    deny = decisions.count("DENY")
    kt = [float(r["K_adj"]) for r in rows]
    print("\n=== Підсумок сесії ===")
    print(f"ALLOW: {allow}, STEP-UP: {stepup}, DENY: {deny}")
    print(f"Середній K_T(adj): {stats.mean(kt):.3f} | Ст.відхилення: {stats.pstdev(kt):.3f}")

# ----------------------------------------
# CLI / entrypoint
# ----------------------------------------

def main():
    parser = argparse.ArgumentParser(
        description="Zero Trust: розрахунок довіри та симуляція сесії"
    )
    parser.add_argument("--scenario", choices=["single", "session"], default="single",
                        help="single — одиничний розрахунок; session — симуляція сесії")
    parser.add_argument("--S", type=float, default=0.8, help="Чутливість ресурсу S(r) ∈ [0,1]")
    parser.add_argument("--dt", type=float, default=15.0, help="Крок часу (хв/ум.од.)")
    parser.add_argument("--steps", type=int, default=10, help="Кількість кроків для симуляції")
    parser.add_argument("--rb", type=str, default="noisy", choices=["stable", "noisy", "suspicious"],
                        help="Профіль поведінкового ризику")
    args = parser.parse_args()

    # Вага ознак та вхідні дані (можеш змінити під свій експеримент)
    # X = [ідентичність, стан пристрою, контекст, поведінка, інше], нормовані [0,1]
    X = [0.90, 0.80, 0.60, 0.70, 0.50]
    weights = ModelWeights(
        beta0=-1.0,
        betas=[1.2, 1.0, 0.8, 0.7, 0.5]
    )

    policy = Policy(
        tau0=0.50, a=0.30, delta=0.10,
        lam=0.02, K0=0.20,
        eta=0.60
    )

    model = ZeroTrustModel(policy=policy, weights=weights)
```

Продовження додатку Б

```
    if args.scenario == "single":
        # Одиничний розрахунок для прикладу з консолі/скріну
        Rb = 0.30
        res = model.evaluate(x=X, dt=args.dt, Rb=Rb, S=args.S)
        print("Zero Trust Simulation Result (single):")
        print_single_eval(res)
    else:
        # Симуляція сесії
        rows = simulate_session(
            model=model,
            x0=X,
            S=args.S,
            steps=args.steps,
            dt=args.dt,
            rb_profile=args.rb
        )
        print("Zero Trust Simulation Result (session):")
        for r in rows:
            print(
                f"t={r['t']:>5.1f} | Rb={r['Rb']:.2f} | x={r['x']} | "
                f"K_T(adj)={r['K_adj']:.3f} | decision={r['decision']}"
            )
        print_session_summary(rows)

if __name__ == "__main__":
    main()
```