КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ

Кафедра радіотехніки та радіоелектронних систем

«На правах рукопису»

Робота допущена до захисту в ЕК
рішенням кафедри радіотехніки та радіоелектронних систем
від _____2026 року, протокол № _____
В.о.завідувача кафедри к.фіз.-мат.наук, доцент
_____ Ігор БЕХ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА
на тему:

«ДИНАМІЧНА МОДЕЛЬ БЕЗПЕКИ З КОНТЕКСТНО-АДАПТИВНОЮ ВЕРИФІКАЦІЄЮ НА РОЗПОДІЛЕНІЙ ТЕСТОВІЙ ІНФРАСТРУКТУРІ»

Виконав:
студент 2-го курсу магістратури
денної форми здобуття освіти
спеціальності 172 Електронні комунікації та радіотехніка
Інформаційна безпека телекомунікаційних систем і мереж
Фалюш Володар Сергійович_____

Науковий керівник:
канд. фіз.-мат. наук, асистент
Котов Михайло Миколайович_____

Рецензент:
с. н. с. відділу науково-архівних фондів Державного
наукового центру захисту культурної спадщини
від техногенних катастроф, к.т.н
Лобузіна Катерина Вілентіївна_____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів без
відповідних посилань
Студент _____ Володар ФАЛЮШ

Київ – 2026

РЕФЕРАТ

Дипломна робота: 51 сторінка, 13 рисунків, 30 джерел.

Ключові слова: Zero Trust, контекстно-орієнтована безпека, динамічна довіра, контроль доступу, HomeLab, Nextcloud, ризик-орієнтована автентифікація, кібербезпека, оцінювання довіри, програмно-апаратний шлюз

Об'єкт дослідження - система контролю доступу до приватної хмарної інфраструктури типу HomeLab.

Мета роботи - розробити та дослідити модель динамічної оцінки довіри мобільного клієнта в системі контролю доступу HomeLab, а також побудувати прототип програмно-апаратного шлюзу, який на основі цієї моделі керує доступом до Nextcloud-сервісу.

Методи дослідження - аналіз наукових публікацій з інформаційної безпеки, моделювання процесів контролю доступу, побудова математичних моделей на основі контекстних параметрів, програмна реалізація та експериментальне тестування прототипу системи в умовах лабораторного середовища.

Результати роботи - розроблено контекстно-орієнтовану модель динамічної оцінки довіри мобільного клієнта в системі контролю доступу до приватної хмарної інфраструктури типу HomeLab та побудовано прототип програмно-апаратного шлюзу, що реалізує цю модель. Запропонована модель враховує геолокаційний, часовий та інтеграційний контексти, формує інтегральний показник довіри й автоматично обирає рівень доступу до мережевого сховища даних. На основі експериментального стенда HomeLab проведено серію сценаріїв тестування, які підтвердили працездатність системи, її здатність адаптивно реагувати на зміну контексту та знижувати ризик несанкціонованого доступу порівняно з традиційними статичними політиками.

ЗМІСТ

РЕФЕРАТ **2**

ЗМІСТ **3**

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ **4**

ВСТУП **5**

1. ОСНОВИ АРХІТЕКТУРИ НУЛЬОВОЇ ДОВІРИ **6**

1.1. Традиційні підходи до забезпечення безпеки локальних мереж **7**

1.2. Концепції Нульової довіри та контекстно-орієнтована безпека **8**

1.3. Моделювання довіри та оцінювання ризику; огляд існуючих моделей **10**

1.4. Шифрування даних та концепція Plausible Deniability **13**

2. ПОСТАНОВКА ЗАДАЧІ ТА ВИМОГИ **17**

2.1. Формалізація наукової проблеми **18**

2.2. Вимоги до архітектури системи **20**

2.3. Критерії оцінки якості системи **22**

3. АРХІТЕКТУРА СИСТЕМИ НУЛЬОВОЇ ДОВІРИ У HOMELAB **24**

3.1. Загальна архітектура **26**

3.2. Компоненти **28**

3.3. Взаємодія компонентів і сценарії доступу **30**

4. МАТЕМАТИЧНА МОДЕЛЬ ДИНАМІЧНОЇ ДОВІРИ **31**

4.1 Контекстні фактори **32**

4.2 Формула коефіцієнта довіри **33**

4.3 Логіка прийняття рішень **34**

5. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОТОТИПУ **36**

5.1 Технічні засоби й ПЗ **36**

5.2 Реалізація компонентів та Сценарії тестування **38**

5.3 Аналіз результатів моделювання **40**

ВИСНОВОК **45**

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ **46**

ДОДАТОК **49**

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

AS - Access Score (Інтегральний показник довіри)

ACL - Access Control List (Список контролю доступу)

API - Application Programming Interface (Програмний інтерфейс застосунків)

IAM - Identity and Access Management (Керування ідентичностями та доступом)

LAN - Local Area Network (Локальна обчислювальна мережа)

MFA - Multi-Factor Authentication (Багатофакторна автентифікація)

TLS - Transport Layer Security (Протокол захисту транспортного рівня)

VPN - Virtual Private Network (Віртуальна приватна мережа)

ZT - Zero Trust (Модель «нульової довіри»)

ZTA - Zero Trust Architecture (Архітектура Zero Trust)

ZTNA - Zero Trust Network Access (Модель доступу до мережі в Zero Trust)

RBAC - Role-Based Access Control (Рольова модель керування доступом)

ABAC - Attribute-Based Access Control (Атрибутивна модель керування доступом)

IdP - Identity Provider (Постачальник ідентичностей (служба автентифікації)

SSO - Single Sign-On (Єдиний вхід до кількох сервісів)

FAR - False Acceptance Rate (Ймовірність хибного допуску)

FRR - False Rejection Rate (Ймовірність хибної відмови)

EER - Equal Error Rate (Точка рівності помилок (FAR = FRR))

REST - Representational State Transfer (Архітектурний стиль веб-сервісів)

JSON - JavaScript Object Notation (Формат обміну структурованими даними)

IP - Internet Protocol (Протокол міжмережевої взаємодії)

DNS - Domain Name System (Система доменних імен)

HomeLab - Домашня лабораторна інфраструктура

Вступ

У сучасних локальних і домашніх мережах дедалі частіше з’являються критично важливі сервіси: хмарні сховища, системи керування «розумним» домом, особисті репозиторії коду, медіасервери тощо. Традиційні статичні політики доступу (списки контролю доступу, фіксовані правила брандмауера, сегментація VLAN) орієнтовані переважно на периметрову модель безпеки «довіряй усередині, не довіряй зовні» і практично не враховують контекст підключення конкретного клієнта [1]. Це призводить або до надмірної «м’якості» політик (ризик компрометації), або до занадто жорстких обмежень, які ускладнюють роботу легітимним користувачам.

Концепція Zero Trust та контекстно-орієнтована безпека пропонують інший підхід: жодному користувачу чи пристрою не довіряють за замовчуванням, а рівень доступу визначається динамічно на основі сукупності факторів - геолокації, часу підключення, стану пристрою, історії взаємодії тощо [2]. Такі рішення активно розвиваються у великих комерційних платформах, однак вони є складними та дорогими для невеликих інфраструктур і домашніх лабораторій, де, тим не менш, зберігаються чутливі дані та особиста інформація користувачів.

Практична потреба полягає в тому, щоб побудувати доступнішу Zero Trust-архітектуру для HomeLab-середовища, яка б поєднувала: локальний хмарний сервіс (наприклад, Nextcloud), шлюз динамічної оцінки довіри та засоби безпечного доступу ззовні без складних VPN-рішень. У такій системі кожна спроба входу мобільного клієнта має супроводжуватися розрахунком коефіцієнта довіри на основі контекстних параметрів і подальшим прийняттям рішення щодо рівня доступу.

1. ОСНОВИ АРХІТЕКТУРИ НУЛЬОВОЇ ДОВІРИ

Сучасні системи контролю доступу в телекомунікаційних мережах і хмарних інфраструктурах розвиваються в умовах постійного зростання кількості користувачів, пристроїв та сервісів, а також суттєвого ускладнення їхніх взаємозв'язків. Традиційні периметрові моделі безпеки, орієнтовані на жорстке розділення «внутрішнього» та «зовнішнього» середовищ, виявляються недостатніми у випадках, коли доступ до ресурсів здійснюється з різних мережевих зон, через проміжні хмарні сервіси та з використанням гетерогенних клієнтських платформ. Це особливо помітно в невеликих інфраструктурах типу HomeLab, де одночасно поєднуються елементи корпоративних архітектур (гіпервізори, віртуальні машини, приватні хмарні сховища) та побутові мережеві рішення.

На цьому фоні активно розвиваються концепції, які розглядають довіру до суб'єктів доступу як динамічну величину, що залежить від контексту сеансу та поточної оцінки ризиків. У моделях класу Нульової довіри жоден користувач або пристрій не вважається апріорі довіреним, а рішення про доступ приймаються на основі багатьох параметрів: геолокації, часу, стану пристрою, історії поведінки та інших ознак. Додатково у багатьох роботах пропонується використовувати числові показники довіри (trust score) як інструмент для побудови ризик-орієнтованих політик керування доступом.

У даній роботі технологічні основи охоплюють три взаємопов'язані напрями. По-перше, це класичні механізми забезпечення безпеки локальних і глобальних мереж: списки контролю доступу (Access Control List, ACL), брандмауери, сегментація мережі, VPN-технології, механізми автентифікації та авторизації на рівні прикладних протоколів. По-друге, це сучасні концепції Нульової Довіри та контекстно-орієнтованої безпеки (Context-Aware Security), які вводять динамічну оцінку довіри й контекстно-орієнтовані правила доступу замість статичних рішень типу «дозволити/заборонити». По-третє, це математичні моделі та програмні засоби, що дозволяють формалізувати процес оцінки довіри, інтегрувати його з інфраструктурою HomeLab та реалізувати прототип шлюзу, здатного застосовувати прийняті рішення до реального мережевого сховища.

1.1. Традиційні підходи до забезпечення безпеки локальних мереж

Історично основою безпеки локальних мереж була периметрово-орієнтована модель, у межах якої вважалося, що внутрішнім вузлам можна довіряти, а зовнішні підключення є потенційно ворожими. Для реалізації цієї моделі застосовувалися класичні засоби: списки контролю доступу (ACL) на маршрутизаторах і комутаторах, статичні правила брандмауерів, сегментація мережі за допомогою VLAN, а також VPN-технології для створення захищених тунелів між віддаленими користувачами та внутрішніми ресурсами [3]. Перевагою такого підходу є простота реалізації та зрозумілість політик: адміністратор один раз налаштовує правила, після чого трафік або пропускається, або блокується на основі адрес, портів і протоколів.

Разом з тим периметрові схеми мають принципові обмеження. Після подолання зовнішнього бар'єра (наприклад, через експлуатацію вразливості сервера, компрометацію облікового запису або помилки конфігурації) зловмисник часто отримує надмірні привілеї всередині мережі, оскільки подальший контроль доступу виконується на основі грубих критеріїв «усередині/зовні» без тонкої диференціації довіри [4]. Крім того, традиційні підходи майже не враховують динамічний контекст - час підключення, географічне розташування, історію пристрою чи аномалії його поведінки. У кращому випадку ці фактори враховуються вручну, наприклад через додаткові правила для певних підмереж або часових інтервалів, що істотно ускладнює адміністрування в міру зростання складності інфраструктури.

У випадку невеликих стендів типу HomeLab традиційні засоби (ACL, VPN, статичні фаєрвол-правила) залишаються важливими базовими інструментами захисту, однак самі по собі вони не розв'язують проблеми гнучкого та адаптивного керування доступом до чутливих ресурсів. Домашні лабораторії часто поєднують

доступ як з локальної мережі, так і з Інтернету, причому з різних пристроїв (персональні комп’ютери, смартфони, планшети), що робить статичні політики надто грубими: або занадто ліберальними, або надмірно жорсткими. Це створює передумови для переходу до моделей, у яких рішення про доступ приймаються не лише на основі належності вузла до певної мережевої зони, а й з урахуванням контексту сеансу та числової оцінки довіри до конкретного клієнта.

1.2. Концепції Нульової довіри та контекстно-орієнтована безпека

Модель Нульової довіри (Zero Trust) сформульована Дж. Кіндервагом у роботах Forrester як відповідь на обмеження традиційних периметрових підходів до захисту, що базуються на принципі «довіряй, але перевіряй» [5]. Її ключова ідея полягає у тезі «ніколи не довіряй, завжди перевіряй», згідно з якою жоден користувач, пристрій чи сервіс не вважаються апріорі довіреними незалежно від їхнього розташування всередині чи поза межами мережевого периметру. Нульова довіра передбачає мінімізацію привілеїв (least privilege), мікросегментацію ресурсів, обов’язкову аутентифікацію й авторизацію для кожного запиту, а також суцільний моніторинг і журналювання трафіку з подальшим аналізом аномалій.

На практиці ці принципи реалізуються у вигляді архітектур Zero Trust Network Access (ZTNA), де рішення про доступ приймаються на рівні окремих застосунків, а не мережевих сегментів, з опорою на ідентичність користувача, стан пристрою, географічне походження запиту та інші контекстні фактори [6]. Ведучі постачальники хмарних сервісів (Google BeyondCorp, Microsoft Azure Conditional Access, Okta Adaptive Authentication, Cloudflare Zero Trust) пропонують комплексні рішення, які інтегрують модель Нульової довіри з корпоративними системами керування ідентичностями (IAM) та каталогами користувачів, дозволяючи будувати політики доступу на основі атрибутів користувача, пристрою і середовища [7-10]. У курсовій роботі ці підходи були розглянуті як референсні для

побудови «легковагового» Zero Trust-рішення в умовах приватної інфраструктури HomeLab.

Доповненням до Zero Trust виступає концепція Context-Aware Security (контекстно-орієнтована безпека), у межах якої рішення про доступ приймаються з урахуванням широкого набору контекстних атрибутів: IP-адреси, геолокації, часу доби, типу та стану пристрою, історії попередніх сесій і поведінкових характеристик користувача. У сучасних системах Identity and Access Management такі підходи реалізуються через механізми адаптивної (ризик-орієнтованої) автентифікації, де спеціалізований «risk engine» обчислює інтегральний ризиковий або довірчий показник та відповідно коригує вимоги до автентифікації (додаткова MFA, пониження рівня доступу, блокування спроби). У цьому проєкті ідеї Context-Aware Security адаптовано у вигляді моделі динамічної оцінки довіри до мобільного клієнта на основі трьох факторів (геолокація, час, цілісність пристрою) з подальшим використанням результату для керування доступом до Nextcloud-сервісу в HomeLab.

Особливістю запропонованого підходу є орієнтація на відкриті та відносно прості для впровадження технології, сумісні з ресурсними обмеженнями домашньої лабораторії: віртуалізаційна платформа Proxmox яка зображена на Рисунку 1.1 та 1.2 для розгортання сервісів, маршрутизатор з підтримкою ACL/Firewall та статичної сегментації, приватне хмарне сховище на базі Nextcloud і власний FastAPI-шлюз для реалізації логіки Нульової довіри та Context-Aware рішень. Такий підхід дозволяє експериментально дослідити можливості сучасних архітектур безпеки у маломасштабному середовищі та продемонструвати, що принципи Нульової довіри і контекстної безпеки можуть бути застосовані не лише у великих корпоративних мережах, але й у домашніх HomeLab-інфраструктурах.

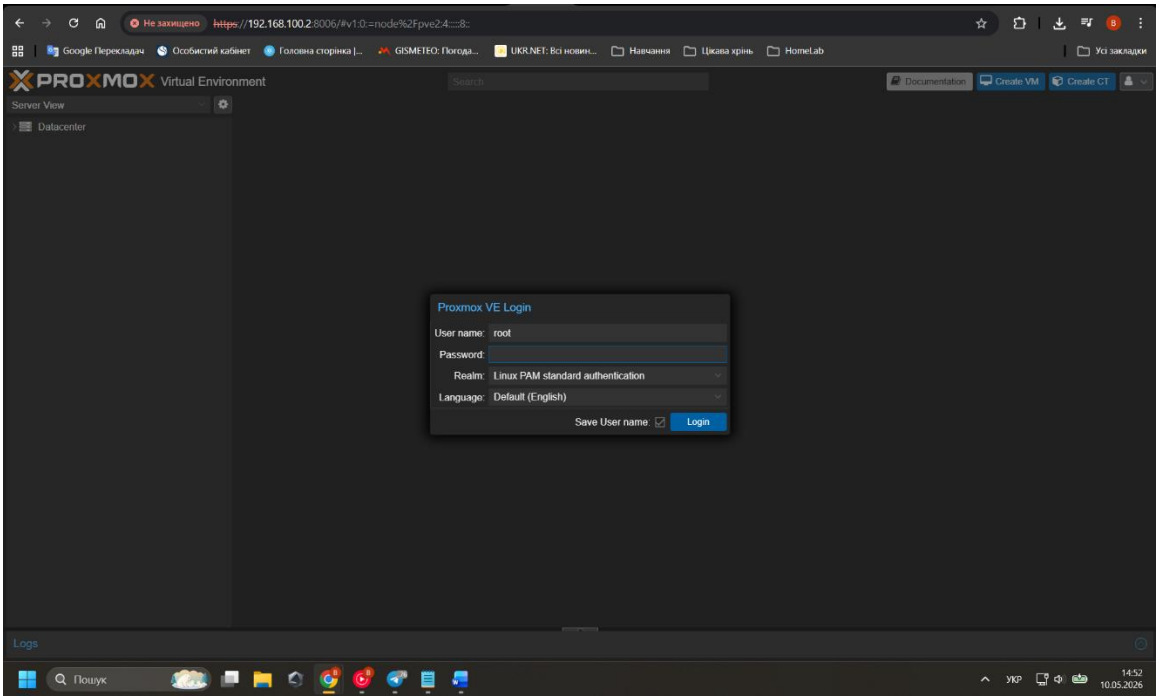

Рис. 1.1 – Вікно логіну у веб інтерфейсі Proxmox

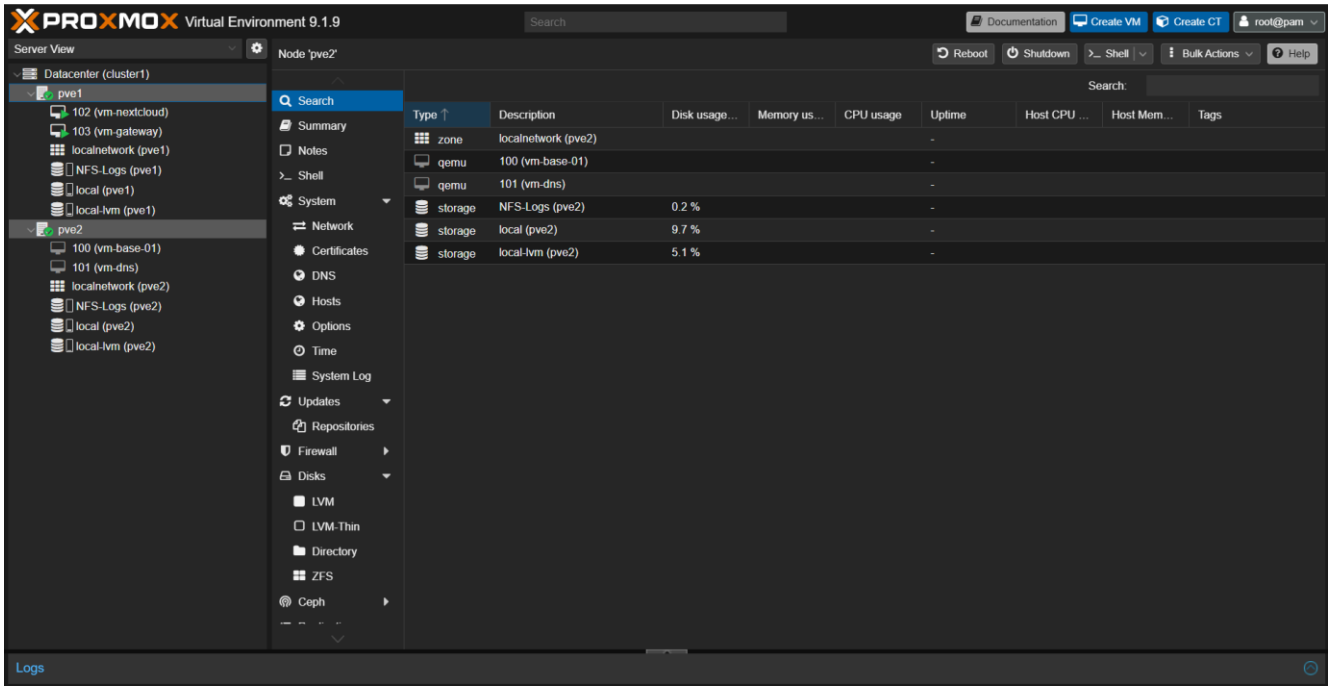

Рис. 1.2 – Вікно інтерфейсу Proxmox з підключеними нодами

1.3. Моделювання довіри та оцінювання ризику; огляд існуючих моделей

Сучасні системи контролю доступу дедалі частіше використовують формальне моделювання довіри й ризику у вигляді числових показників (trust score,

risk score), які узагальнюють інформацію про контекст доступу в одну метрику для прийняття рішень [11]. У загальному випадку довіра розглядається як функція від набору факторів $f_1, f_2, \dots, f_n,$, що описують властивості користувача, пристрою, мережі та операційного контексту, тоді як ризик інтерпретується як ймовірність та/або очікуваний збиток від потенційно шкідливої дії за наявного стану цих факторів. У роботах, присвячених адаптивній автентифікації та risk-based access control, типово застосовуються агреговані показники, які використовуються для класифікації сесій за рівнями безпеки з подальшим застосуванням відповідних політик (дозволити, вимагати підвищену автентифікацію, заблокувати) [12].

Точність таких моделей часто характеризується через метрики, запозичені з теорії розпізнавання образів та біометричної автентифікації: показник хибного прийняття (False Acceptance Rate, FAR), що відображає частку несанкціонованих спроб, помилково класифікованих як легітимні, показник хибного відхилення (False Rejection Rate, FRR), який описує частку правомірних спроб, помилково відхилених системою, а також точку рівності помилок (Equal Error Rate, EER), у якій FAR і FRR збігаються [13]. Вибір порогів для розмежування рівнів довіри (наприклад, інтервали для рішень «повний доступ», «обмежений доступ», «блокування») безпосередньо впливає на баланс між зручністю користувача та безпекою системи, і в практичних реалізаціях зазвичай здійснюється експериментально на основі аналізу накопичених журналів подій.

У промислових рішеннях Нульової довіри та адаптивної автентифікації часто застосовуються складні ризикові рушії, які комбінують експертні правила, статистичні моделі й машинне навчання для оцінки ризику в реальному часі, враховуючи поведінкові, контекстні та історичні ознаки. Типові фактори включають географічне розташування користувача, відхилення у часових шаблонах входу, відомості про пристрій (наявність шифрування, версія ОС, статус комплаєнсу), «репутацію» IP-адреси, історію попередніх сесій та наявну розвідінформацію про загрози [14,24]. На основі цих даних обчислюється узагальнений ризиковий бал, який визначає, чи буде застосована стандартна схема доступу, посилена автентифікація (step-up MFA) або повна відмова в доступі.

У цій роботі, спираючись на результати курсового дослідження, для HomeLab-інфраструктури обрано свідомо спрощену, але прозору модель довіри у вигляді зваженої суми нормованих контекстних факторів. Інтегральний показник довіри $Tscore$ задається формулою :

$$Tscore = wgeo \cdot fgeo + wtime \cdot ftime + wint \cdot fint,$$

де $fgeo, ftime, fint \in [0; 1]$ - числові оцінки геолокаційного, часовго та «інтеграційного» (цілісність пристрою й історія підключень) факторів відповідно, а $wgeo, wtime, wint$ - невід'ємні вагові коефіцієнти, що сумарно дорівнюють одиниці. Така лінійна модель дає змогу явно контролювати внесок кожного фактору, легко інтерпретується адміністраторами та добре підходить для експериментального середовища HomeLab, де важлива не лише ефективність, але й пояснюваність прийнятих рішень.

Подальше перетворення $Tscore$ у дискретні рішення виконується за допомогою двох порогових значень T_{low} та T_{high}, що визначають три режими доступу: «повний доступ» (Full Access) при $Tscore \geq Thigh$, «обмежений доступ» (Limited Access) при $Tlow \leq Tscore < Thigh$ та «блокування» (Blocked Access) при $T_{score} < T_{low}$. В імплементації на базі FastAPI-шлюзу ці пороги, разом з вагами факторів, зберігаються у таблиці TrustRule СУБД, а для кожного запиту формується запис у таблиці AccessEvent, що містить обчислені значення часткових оцінок, інтегральний показник, прийняте рішення та текстове пояснення. Проведений у курсовій порівняльний аналіз показав, що така модель дозволяє досягти більш гнучкого балансу між безпекою і зручністю порівняно зі статичними ACL та простими контекстними правилами, зокрема у сценаріях зміни Wi-Fi-мережі, появи нового пристрою чи доступу з нетипового географічного регіону.

1.4. Шифрування даних та концепція Plausible Deniability

Поняття plausible deniability (правдоподібне заперечення причетності до даних) у криптографії використовується для опису схем шифрування, за яких наявність певних конфіденційних даних не може бути надійно доведена третім сторонам, навіть якщо їм відомо про факт використання шифрування. На відміну від звичайного повнодискового шифрування, де сам факт існування зашифрованого тома або розділу є очевидним, системи з підтримкою plausible deniability дозволяють приховати чутливі томи всередині інших, менш критичних, або ж моделювати ситуацію, коли користувач має лише «загальнодоступні» дані. Такий підхід застосовується, зокрема, у реалізаціях на основі LUKS (Linux Unified Key Setup) та VeraCrypt, де підтримуються багатошарові схеми шифрування, приховані томи й гнучка система ключів доступу [15-16].

У середовищі GNU/Linux LUKS є де-факто стандартом повнодискового шифрування, який забезпечує надійний захист даних за допомогою сучасних криптографічних примітивів та механізмів керування ключами, включно з можливістю використання декількох ключових слотів для різних категорій користувачів. VeraCrypt, у свою чергу, розвиває ідеї проекту TrueCrypt та прямо підтримує режими plausible deniability, дозволяючи створювати приховані томи й організовувати конфігурації, в яких «зовнішній» пароль відкриває малочутливі дані, а «прихований» - критично важливу інформацію. У поєднанні з протоколами захищеного транспорту (TLS, HTTPS) це створює багаторівневий захист: навіть у разі фізичного доступу до накопичувача зловмисник не може довести існування додаткових зашифрованих об’ємів понад ті, які йому було продемонстровано.

У контексті HomeLab-інфраструктури, що розглядається в цій роботі, застосування концепції plausible deniability доповнює динамічну модель довіри і Zero Trust-архітектуру, але працює на іншому рівні - захисті даних «у спокої» Критично важливі об’єми, на яких розміщується приватне хмарне сховище (Nextcloud) та інші чутливі сервіси, можуть бути розгорнуті на зашифрованих томах LUKS або VeraCrypt, при цьому логіка динамічної довіри та

контекстно-орієнтованого доступу відповідає за контроль з’єднань «у русі» (in transit). Таким чином, навіть у разі компрометації мережевого периметра або частини компонентів Zero Trust-системи зловмисник не отримує автоматичного доступу до вмісту зашифрованих томів, а користувач зберігає можливість правдоподібно заперечувати існування додаткових чутливих даних. Інтерфейс Nextcloud представлений на Рисунку 1.3 та 1.4, і в даному випадку використовується лише як зручний веб інтерфейс для доступу та керування файлами, а також відображення уже налаштованих просторів.

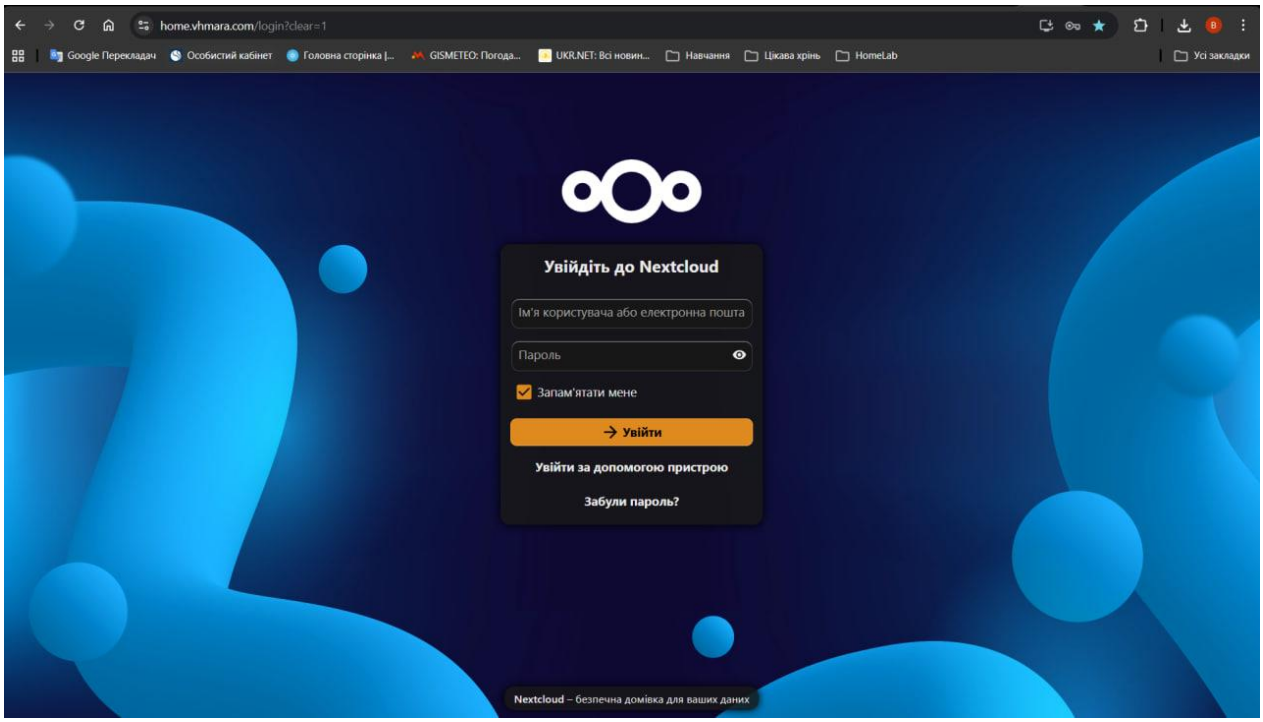

Рис. 1.3 – Вікно логіну у Nextcloud

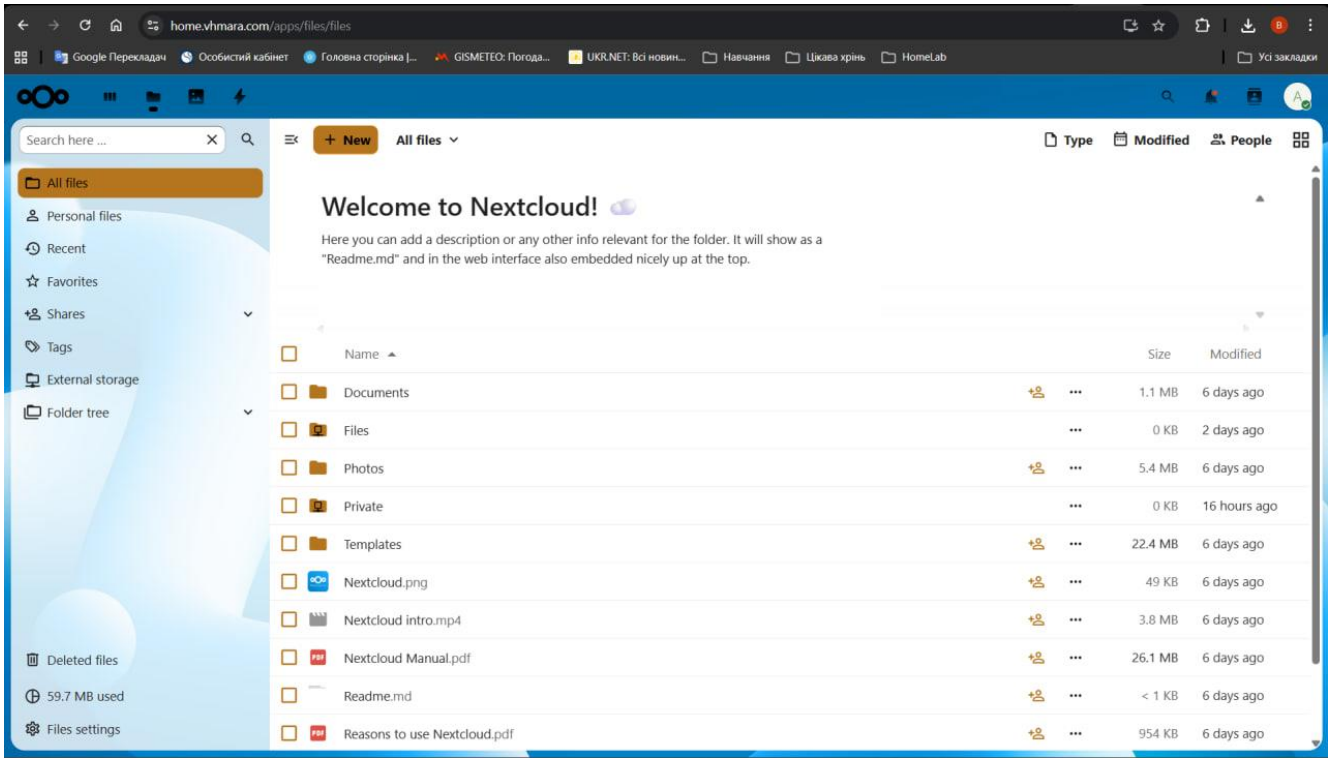

Рис. 1.4 – Відображення папок та файлів у системі Nextcloude

Саме ж сховище знаходиться фізично на робочій станції Dell T3610 з ОС TrueNas, яка розрахована для створення саме NAS сховищ, і налаштування мережевого простору папок, їхніх розмірів та доступів відбувається саме тут. Інтерфейс TrueNas зображений на рисунку 1.5 та 1.6.

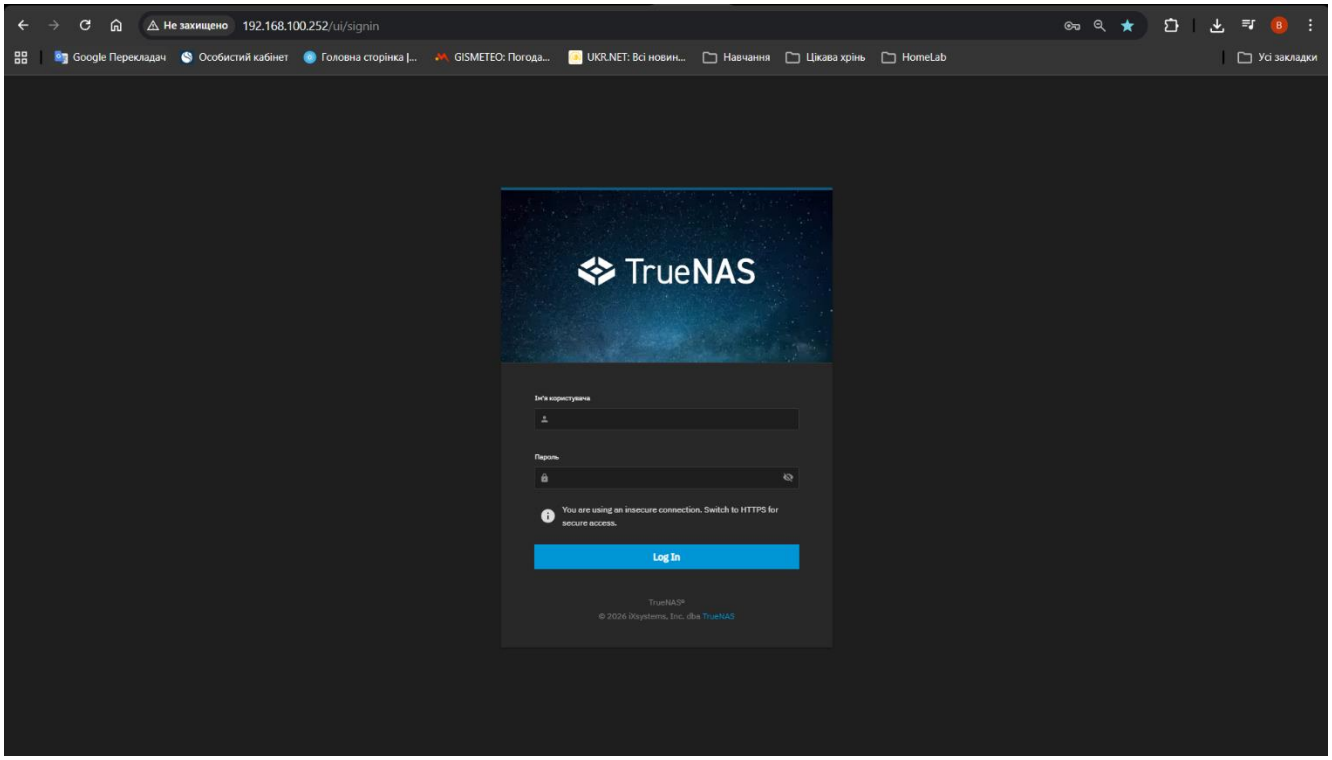

Рис. 1.5 – логін у TrueNAS після увімкнення системи та переходоі за локальним IP комп'ютера

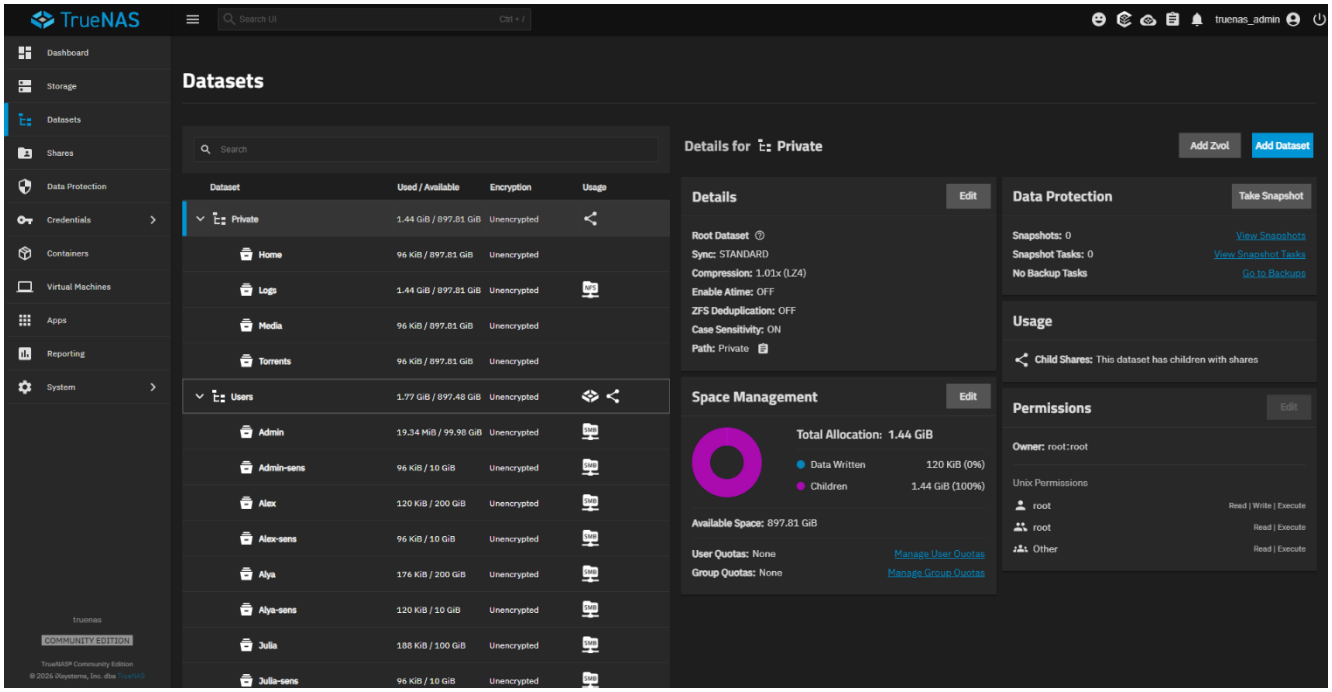

Рис. 1.6. – Інтерфейс системи, з відображенням налаштованих просторів для користувачів

На даному фото зображено інтерфейс TrueNAS, і налаштовані простори для кожного користувача, з обмеженнями по місцю зберігання яке виділено та прописано для кожного.

2. ПОСТАНОВКА ЗАДАЧІ ТА ВИМОГИ

Інфраструктури типу HomeLab, у яких поєднуються віртуалізаційні платформи, приватні хмарні сховища, мережеві служби та віддалений доступ з різних клієнтських пристроїв, за рівнем складності дедалі більше наближаються до малих корпоративних мереж, але при цьому експлуатуються окремими користувачами або невеликими колективами без суттєвих фінансових ресурсів. У таких умовах традиційні підходи до захисту, засновані на статичних ACL, VPN-тунелях та фіксованих правилах брандмауера, виявляються недостатніми для врахування динамічного контексту доступу (геолокація, час доби, історія пристрою, аномальна активність) і не дозволяють реалізувати сучасні принципи Нульової довіри та контекстно-орієнтованої безпеки.

Сучасні промислові платформи Zero Trust (Google BeyondCorp, Microsoft Azure Conditional Access, Okta Adaptive Authentication, Cloudflare Zero Trust) пропонують потужні механізми динамічного керування доступом, однак вони орієнтовані насамперед на великі організації, інтегровані з корпоративними IAM-системами та потребують значних ресурсів[6-9]. Для власника HomeLab така інфраструктура є надмірною як у сенсі вартості, так і у вимогах до адміністрування, що створює науково-прикладну нішу для розроблення «легковагових» моделей динамічної довіри та їхньої реалізації на базі відкритого програмного забезпечення. З огляду на це, робота спрямована на побудову та дослідження моделі динамічної оцінки довіри до мобільного клієнта в HomeLab-інфраструктурі, яка використовує обмежений, але інформативний набір контекстних факторів та інтегрується з існуючими компонентами типового домашнього стенда (Proxmox, маршрутизатор із ACL/Firewall, Nextcloud)[15-16].

Постановка задачі передбачає, по-перше, формалізацію наукової проблеми у вигляді задачі оцінювання довіри, де кожній сесії доступу ставиться у відповідність числовий показник $T_{score} \in [0; 1]$, що відображає сукупний рівень довіри до користувача й його пристрою за поточного контексту. По-друге, необхідно визначити вимоги до архітектури системи, яка реалізує цю модель у вигляді програмно-апаратного шлюзу, здатного приймати рішення про рівень доступу до

приватного хмарного сховища (Nextcloud) на основі значення T_{score} та передавати ці рішення через стандартні API (Provisioning API Nextcloud, механізми груп доступу). По-третє, потрібно задати критерії оцінки якості системи - зокрема, здатність зменшувати ймовірність несанкціонованого доступу при прийнятному рівні зручності для легітимних користувачів, що може бути охарактеризовано через показники FAR/FRR, поведінку в типових сценаріях і порівняння зі статичними політиками ACL.

У результаті постановка задачі включає три взаємопов'язані аспекти: теоретичний (розроблення формальної моделі довіри на основі контекстних факторів), архітектурний (розроблення структури Zero Trust HomeLab з інтегрованим шлюзом динамічної довіри) та експериментальний (реалізація прототипу на базі FastAPI, бази даних журналів подій та інтеграції з Nextcloud для подальшого тестування й аналізу). Така багаторівнева постановка забезпечує як наукову новизну у частині моделювання довіри для маломасштабних хмарних інфраструктур, так і практичну значущість у вигляді працюючого рішення, придатного для застосування в реальних HomeLab-сценаріях.

2.1. Формалізація наукової проблеми

Формально наукову проблему, що розв'язується в роботі, можна описати як задачу побудови моделі динамічної довіри до клієнта системи контролю доступу, яка відображає багатовимірний контекст сесії у скалярну величину $\boldsymbol{T_{score}}$ та використовується для прийняття рішень про рівень доступу до ресурсів приватної хмарної інфраструктури. Нехай вектор контекстних факторів має вигляд

$$\mathbf{f} = (\boldsymbol{f_{geo}}, \boldsymbol{f_{time}}, \boldsymbol{f_{int}}, \dots),$$

де $\boldsymbol{f_{geo}}$ характеризує геолокаційний контекст (тип мережі, країна/регіон за IP, приналежність до відомих безпечних зон), $\boldsymbol{f_{time}}$ - часові характеристики (день тижня, часовий інтервал, відповідність типовому графіку користувача), а $\boldsymbol{f_{int}}$ -

інтеграційний фактор, що відображає стан і історію пристрою (новий/відомий пристрій, зміна підмережі IP, результати перевірок цілісності).

Потрібно побудувати функцію агрегування

$$\boldsymbol{T_{score}} = \boldsymbol{F}(\mathbf{f}),$$

яка відображає цей вектор у інтервал $[\mathbf{0}; \mathbf{1}]$ так, щоб значення, близькі до одиниці, відповідали високому рівню довіри (низькому ризику), а значення, близькі до нуля, - низькому рівню довіри (високому ризику). У роботі, спираючись на результати курсового дослідження, обирається зважена лінійна модель

$$\boldsymbol{T_{score}} = \boldsymbol{w_{geo}f_{geo}} + \boldsymbol{w_{time}f_{time}} + \boldsymbol{w_{int}f_{int}},$$

де коефіцієнти $\boldsymbol{w_{geo}, w_{time}, w_{int}} \geq \mathbf{0}$ та $\boldsymbol{w_{geo}} + \boldsymbol{w_{time}} + \boldsymbol{w_{int}} = \mathbf{1}$, що забезпечує прозорість та інтерпретованість внеску кожного фактора в загальний рівень довіри. Таке формулювання дозволяє явно налаштовувати чутливість моделі до зміни окремих контекстних параметрів і є узгодженим з підходами до побудови trust score у сучасних системах адаптивної автентифікації та risk-based access control[2,17].

Друга складова формалізації полягає у задання відображення значення $\boldsymbol{T_{score}}$ у кінцеву множину рішень щодо доступу $\{\boldsymbol{Full}, \boldsymbol{Limited}, \boldsymbol{Blocked}\}$, що реалізується за допомогою двох порогів $\boldsymbol{T_{low}}$ та $\boldsymbol{T_{high}}$ з умовами

$\boldsymbol{T_{score}} \geq \boldsymbol{T_{high}}$ - повний доступ до сервісів (наприклад, монтування чутливих томів Nextcloud або доступ до групи hightrust);

$\boldsymbol{T_{low}} \leq \boldsymbol{T_{score}} < \boldsymbol{T_{high}}$ - обмежений доступ (тільки базовий функціонал, без чутливих ресурсів);

$\boldsymbol{T_{score}} < \boldsymbol{T_{low}}$ - блокування запиту або відмова у доступі.

Третя складова пов’язана з інтеграцією формальної моделі в реальну систему Zero Trust HomeLab, де шлюз на базі FastAPI отримує від клієнта необхідні ідентифікаційні та контекстні дані, обчислює $\boldsymbol{T_{score}}$, приймає рішення про рівень доступу та, за потреби, викликає Provisioning API Nextcloud для перевірки існування користувача й керування його належністю до групи підвищеної довіри. Важливою вимогою є реалізація політики fail-closed: у випадку помилки взаємодії з зовнішнім сервісом, некоректних або неповних даних, а також перевищення

частоти запитів (rate limiting) система повинна відхиляти запит або встановлювати $\boldsymbol{T_{score} = 0}$, а не надавати доступ за замовчуванням.

Таким чином, формалізація наукової проблеми зводиться до побудови цілком визначеної математичної моделі динамічної довіри з явними параметрами (ваги та пороги), визначення процедур обчислення часткових факторів на основі реальних мережевих даних у HomeLab та специфікації алгоритму прийняття рішень у шлюзі доступу, який є сумісним з існуючою інфраструктурою (Nextcloud, Proxmox, маршрутизатор з ACL/Firewall). Це створює основу для подальшого аналізу властивостей моделі, експериментальних досліджень на лабораторному стенді та порівняння її поведінки з традиційними підходами до контролю доступу в локальних мережах.

2.2. Вимоги до архітектури системи

Архітектура системи контролю доступу для HomeLab-інфраструктури має забезпечувати реалізацію моделі динамічної довіри в умовах обмежених апаратних ресурсів, з використанням переважно відкритого програмного забезпечення та без залежності від комерційних Zero Trust-платформ. З огляду на це, до архітектури формулюються такі ключові функціональні вимоги:

- наявність окремого програмного компонента-шлюзу (policy engine), який приймає запити від клієнтів, збирає контекстні дані (IP-адреса, приблизна країна, час, інформація про пристрій) та реалізує алгоритм обчислення інтегрального показника довіри T_{score};
- інтеграція шлюзу з приватним хмарним сховищем (Nextcloud) через офіційний Provisioning API для перевірки існування облікових записів та керування належністю користувача до груп підвищеної довіри (наприклад, hightrust);

- ведення централізованого журналу подій доступу в реляційній базі даних (таблиця AccessEvent), де фіксуються значення часткових факторів, розрахований T_{score}, ухвалене рішення, рівень доступу й текстове пояснення.

Нефункціональні вимоги охоплюють, по-перше, вимогу до відкритості та перевірюваності реалізації: ключові компоненти мають будуватися на базі Linux, Proxmox VE, відкритих маршрутизаторних платформ із підтримкою ACL/Firewall, приватного хмарного сховища Nextcloud та веб-фреймворку FastAPI з ORM-шаром SQLAlchemy [15,18-20]. По-друге, архітектура повинна бути модульною й розширюваною, щоб додавання нових контекстних факторів (наприклад, перевірки Play Integrity API, поведінкових ознак чи даних про репутацію IP-адреси) потребувало мінімальних змін у загальній структурі системи та не призводило до її повної реконфігурації [21]. По-третє, важливою є вимога до прозорості прийняття рішень: результати обчислень мають бути придатними для інтерпретації адміністратором, що досягається завдяки збереженню не лише інтегрального T_{score}, а й компонентних оцінок $f_{geo}, f_{time}, f_{int}$ та текстового пояснення причини рішення в журналі подій.

З технічної точки зору архітектура повинна включати: віртуалізаційний вузол (Proxmox) для розгортання віртуальних машин і контейнерів; маршрутизатор/фаєрвол (наприклад, MikroTik RouterOS чи OpenWrt), що реалізує базову сегментацію мережі й перенаправлення трафіку до шлюзу; сам шлюз на базі FastAPI, який реалізує REST-інтерфейс api/access-check; сервер бази даних з таблицями User, Device, TrustRule, AccessEvent; а також екземпляр Nextcloud, з яким шлюз взаємодіє через Provisioning API для зміни груп доступу і, за потреби, застосування сценаріїв на кшталт автоматичного додавання/видалення користувача до/з групи hightrust залежно від рішення [22-23]. Додатково накладається вимога на дотримання принципу fail-closed: у разі недоступності Nextcloud-API, помилки розбору відповіді або перевищення ліміту спроб за одиницю часу шлюз повинен відмовляти в доступі (встановлювати $T_{score} = 0$ і рішення blocked або ratelimited), а не надавати доступ за замовчуванням [24].

2.3. Критерії оцінки якості системи

Оцінювання якості запропонованої моделі динамічної довіри та її реалізації в архітектурі Zero Trust HomeLab має ґрунтуватися як на кількісних, так і на якісних критеріях, що відображають здатність системи зменшувати ризик несанкціонованого доступу без надмірного погіршення зручності роботи легітимних користувачів. На рівні математичної моделі важливим є аналіз балансу між показником хибного прийняття (False Acceptance Rate, FAR) та хибного відхилення (False Rejection Rate, FRR), а також пошук таких значень порогів T_{low} і T_{high}, які забезпечують прийнятну точку рівності помилок (Equal Error Rate, EER) для обраних сценаріїв використання [12]. У практичних умовах HomeLab це досягається не стільки строгим статистичним оцінюванням на великих вибірках, скільки аналізом поведінки системи в репрезентативному наборі сценаріїв доступу (типові робочі входи, нові пристрої, нетипові країни, нічний час тощо), з подальшим налаштуванням вагових коефіцієнтів та порогів.

Другу групу критеріїв становить порівняння рішень, які приймає запропонована модель, із рішеннями базових систем контролю доступу, що використовуються як «бенчмарк» - насамперед статичні ACL та проста контекстно-орієнтована модель без інтегрального показника T_{score}. У курсовій роботі було наведено таблиці, де для низки сценаріїв (легітимний доступ з домашньої мережі, підміна Wi-Fi-мережі, доступ через VPN, спроби з IP-адреси атакувальника) порівнювалися результати трьох підходів: Static ACL, Simple Context та запропонованої моделі з T_{score}. Аналіз показав, що статичні ACL часто або надмірно ліберальні (пропускають підозрілий доступ при збігу IP-адреси чи підмережі), або надто консервативні (блокують легітимні сценарії на кшталт доступу через VPN), у той час як запропонована модель здатна розрізняти ці ситуації завдяки врахуванню сукупності факторів GEO, TIME та INTEGRITY.

Третій важливий критерій стосується експлуатаційних властивостей системи: прозорість і пояснюваність рішень, простота налаштування та

розширюваність. Журнал подій AccessEvent, у якому зберігаються як часткові оцінки факторів, так і інтегральний T_{score}, текстове пояснення (reason) та рішення (decision, accesslevel), дозволяє адміністратору ретроспективно аналізувати випадки як дозволеного, так і заблокованого доступу, виявляти помилки налаштувань і коригувати ваги та пороги без модифікації коду шлюзу. До якісних критеріїв також належать відповідність принципу fail-closed (жодний збій зовнішніх компонентів не повинен призводити до неконтрольованого надання доступу), коректність обробки аномальних режимів (rate limiting для спроб brute-force, відхилення неіснуючих облікових записів, некоректні або відсутні контекстні дані) і можливість поетапного розширення моделі довіри шляхом додавання нових факторів та типів подій. Сукупність цих критеріїв дозволяє комплексно оцінити ефективність запропонованої системи порівняно з традиційними підходами до контролю доступу в локальних і гібридних мережах.

3. АРХІТЕКТУРА СИСТЕМИ НУЛЬОВОЇ ДОВІРИ У HOMELAB

Побудова Zero Trust-архітектури в умовах приватної лабораторної інфраструктури типу HomeLab потребує адаптації корпоративних підходів до масштабів та обмежень одного або кількох домашніх серверів. На відміну від великих організацій, де Zero Trust інтегрується з розгалуженими IAM-системами, каталогами користувачів та хмарними сервісами, у середовищі HomeLab усі ключові компоненти (віртуалізація, маршрутизація, хмарне сховище, шлюз доступу) зазвичай адмініструються однією особою й розміщуються в межах локальної мережі або за одним публічним IP-адресним простором. Це зумовлює необхідність спрощення архітектури при збереженні основних принципів Нульової довіри: «ніколи не довіряй, завжди перевіряй», мінімізація привілеїв, мікросегментація та обов'язкова перевірка кожного запиту до ресурсу.

У запропонованій у роботі архітектурі Zero Trust HomeLab ці принципи реалізовані через виділення окремого програмного шлюзу, який виконує роль policy engine та є єдиною точкою прийняття рішень щодо доступу до приватного хмарного сховища Nextcloud й інших сервісів. Усі запити від клієнтів (мобільних пристроїв, робочих станцій) до захищених ресурсів проходять через цей шлюз, де на основі контекстних параметрів (IP-адреса та геолокація, часовий інтервал, історія пристрою) обчислюється інтегральний показник довіри T_{score}, а далі ухвалюється рішення про рівень доступу (повний, обмежений, блокування). Інтерфейс цього шлюзу зображено на Рисунку 3.1. Сам інтерфейс отримав назву zt-gateway, де zt - означає Zero Trust і відображає усі зазначені дані, та повертає кінцеве число після обрахунку, й пояснює які ваги використались. Самі ресурси розгорнуті на віртуалізаційній платформі Proxmox та додатково захищені механізмами повнодискового шифрування, що відповідає принципу багаторівневого захисту. Розрахунок по наданому доступу та їх можливості показано на Рисунку 3.2.

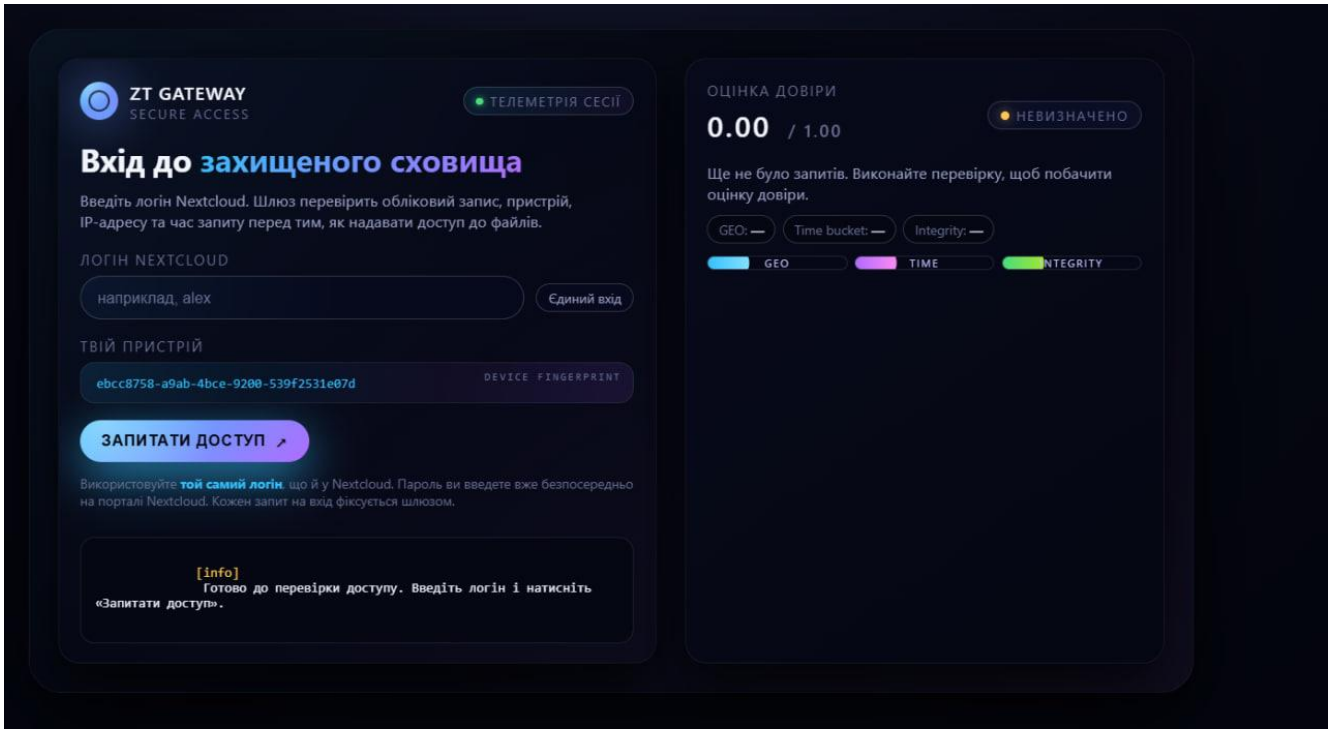

Рис. 3.1 – Веб інтерфейс шлюзу для прорахунку коефіцієнта довіри

Рис. 3.2 - Сценарії рішень

Структурно Zero Trust HomeLab орієнтується на відомі еталонні моделі Zero Trust Network Access, у яких виділяються чотири ролі: клієнтський пристрій (User/Device), система ідентифікації та обліку (Identity Store), рушій політик (Policy Engine) та захищені ресурси (Protected Resources) [10]. У даній роботі ці ролі реалізуються відповідно: мобільним клієнтом з унікальним відбитком пристрою; системою керування обліковими записами Nextcloud, що виступає джерелом істини щодо користувачів; шлюзом на базі FastAPI з інтегрованою моделлю динамічної довіри; а також приватним хмарним сховищем і додатковими сервісами HomeLab, доступ до яких опосередковується через рішення шлюзу. Такий підхід дозволяє відтворити ключові властивості Нульової довіри у маломасштабному

середовищі та створює основу для подальшого розширення інфраструктури без кардинальної перебудови архітектури.

3.1. Загальна архітектура

Загальна архітектура Zero Trust HomeLab має багатошарову структуру, у якій фізичний рівень (домашній сервер, маршрутизатор, комутатор) поєднується з віртуалізаційним середовищем Proxmox, мережевою інфраструктурою з сегментацією, програмним шлюзом динамічної довіри та рівнем прикладних сервісів, зокрема Nextcloud. На мережевому рівні маршрутизатор (наприклад, MikroTik RouterOS чи OpenWrt) виконує функції базового фільтрування трафіку, реалізує NAT або проброс портів до шлюзу, а також за потреби забезпечує VLAN-сегментацію для відокремлення HomeLab-сегмента від інших підмереж [22]. На віртуалізаційному рівні Proxmox розгортає віртуальні машини або контейнери, у яких працюють Nextcloud-сервер, база даних журналів подій та сервіс шлюзу доступу.

Центральним елементом архітектури є веб-шлюз, реалізований за допомогою FastAPI на Python, який надає REST-інтерфейс POST /api/access-check для клієнтів. Кожен запит містить ім’я користувача, відбиток пристрою та, за потреби, IP-адресу; додатково шлюз самостійно визначає IP клієнта й розв’язує його до приблизної країни та регіону за допомогою локальної бази GeoLite2-City (через бібліотеку geoip2) [23]. Для оцінки геолокаційного фактора GEO використовується функція compute_geoscore, яка надає максимальний бал підмережі HomeLab (наприклад, 192.168.100.0/24) та країні UA, проміжні значення для «дружніх» країн (PL, DE, CZ, SK) і знижені значення для невідомих або віддалених регіонів. Часовий фактор TIME визначається функцією classify_time_bucket_and_score, що класифікує момент запиту як «workhours», «evening», «night» або «weekend» та

присвоює відповідні бали залежно від того, наскільки часовий інтервал є типовим для легітимної активності.

Інтеграційний фактор INTEGRITY обчислюється на основі історії пристрою в таблиці Device, де фіксуються перша поява відбитка, останній час активності та попередня IP-адреса. Якщо пристрій новий (вік запису менший за 24 години) або спостерігається зміна підмережі IP-адреси порівняно з попередніми сесіями, шлюз позначає статус як NEW_DEVICE або SUSPICIOUS_IP_CHANGE відповідно і знижує значення відповідного фактора, тоді як для відомих пристроїв у межах тієї ж підмережі виставляється максимальний бал. Далі активне правило довіри TrustRule (що містить ваги факторів та пороги T_{low}, T_{high}) завантажується з таблиці БД, після чого обчислюється інтегральний показник довіри trustscore і на його основі функція decide_access_rule визначає рішення та рівень доступу normal, sensitive, none, або ж Blocked, Limited і Full, як показано на рисунку 3.3

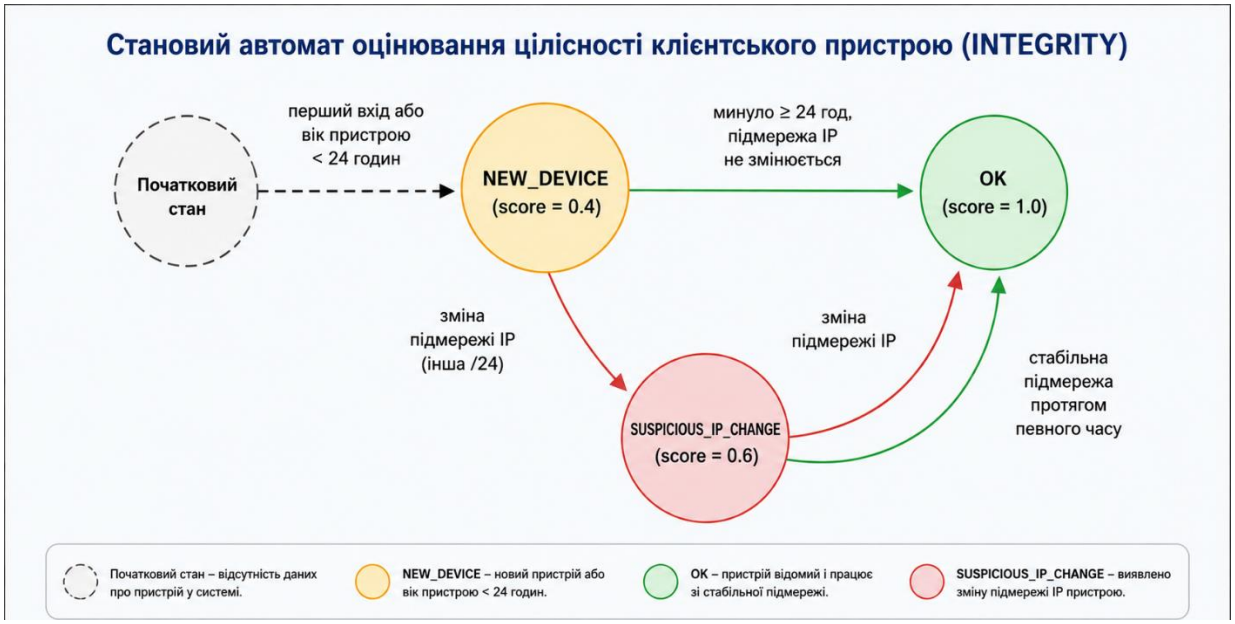

Рис. 3.3 Модель коефіцієнта довіри Tscore

Перед застосуванням рішення шлюз додатково виконує дві перевірки, критичні для безпеки: rate limiting та валідацію облікового запису через Nextcloud Provisioning API. Механізм rate limiting рахує кількість спроб входу з тієї самої IP-адреси за короткий часовий інтервал (наприклад, 10 секунд) та, у разі перевищення порогового значення, примусово встановлює trustscore = 0 і рішення ratelimited, не звертаючись до зовнішнього сервісу, що ефективно блокує

атаки перебору паролів. Перевірка існування користувача (nc_user_exists) виконується через OCS-інтерфейс Nextcloud; якщо користувача не існує або відповідь сервісу некоректна, застосовується політика fail-closed із рішенням invalidusername і забороною доступу. У випадку позитивного рішення й високого рівня довіри шлюз може автоматично додати користувача до групи hightrust, а при зниженні довіри - вилучити з цієї групи, тим самим реалізуючи динамічне керування доступом до чутливих ресурсів Nextcloud на основі моделі T_{score}.

Усі події доступу фіксуються в таблиці AccessEvent зі збереженням ідентифікаторів користувача й пристрою, IP-адреси, країни та регіону, часової категорії, статусу інтеграційного фактора, числових значень часткових оцінок і інтегрального trustscore, прийнятого рішення та логічного пояснення (reason). Це забезпечує не лише поточну роботу системи, але й можливість подальшого аналізу, налаштування ваг і порогів, а також розширення моделі за рахунок додаткових факторів, що відповідає загальним рекомендаціям щодо розвитку Zero Trust-архітектур у напрямку дедалі більш гнучких і контекстно-орієнтованих політик.

3.2. Компоненти

Першим ключовим компонентом є приватний хмарний файловий сервіс, який виступає основним цільовим ресурсом системи. Він забезпечує зберігання, синхронізацію та спільний доступ до файлів, має веб-інтерфейс і клієнтські застосунки, а також підтримує базові механізми керування правами доступу користувачів. Використання саме такого типу сервісу є доцільним, оскільки він концентрує в собі критичні дані користувача, має чіткий HTTP(S)-інтерфейс та добре підходить як «полігон» для дослідження політик доступу.

Другим компонентом є шлюз динамічної довіри, реалізований як веб-служба на базі високопродуктивного фреймворку для побудови REST-API, зокрема FastAPI[18]. Обрання FastAPI обумовлене його асинхронною природою, зручною моделлю опису маршрутів та інтеграцією з Python, що спрощує реалізацію математичної моделі trust-scoring, роботу з базами даних і зовнішніми сервісами. У межах шлюзу реалізовані такі функції: приймання запитів від клієнтів, збір і нормалізація контекстних факторів, обчислення показника довіри, ведення журналів подій, взаємодія з файловим сервісом та Zero Trust-інфраструктурою.

Третім компонентом виступає Zero Trust-інфраструктура тунелювання та публікації додатків, яка забезпечує безпечний доступ із глобальної мережі до внутрішніх веб-ресурсів без прямого відкриття портів на маршрутизаторі. Сервіси класу Cloudflare Tunnel/Access дозволяють організувати стійкий зашифрований тунель від домашнього сервера до периферійної точки присутності провайдера, а також накладати додаткові політики автентифікації й авторизації на рівні периметра, наприклад вимагати аутентифікації через зовнішній Identity-провайдер перед переспрямуванням трафіку до шлюзу [25]. Такий підхід зменшує поверхню атаки та суттєво спрощує мережеву конфігурацію HomeLab у порівнянні з класичними схемами порт-форвардингу.

Додатковими елементами архітектури є сервер віртуалізації, де розгорнуті віртуальні машини з файловим сервісом і шлюзом; система збирання та зберігання логів; а також, за потреби, внутрішній DNS-сервіс для зручного іменування ресурсів. Разом ці компоненти утворюють замкнену екосистему, в якій усі критичні елементи розташовані під контролем адміністратора HomeLab.

3.3. Взаємодія компонентів і сценарії доступу

У типового сценарію доступу мобільний клієнт спочатку звертається до публічного доменного імені, пов’язаного із Zero Trust-тунелем. Трафік шифрується та надходить до зовнішнього вузла тунелювання, де застосовуються базові політики периметрового контролю, після чого запит переспрямовується до шлюзу динамічної довіри у внутрішній мережі. На цьому етапі шлюз ініціює процедуру автентифікації користувача й паралельно збирає контекстні дані: параметри IP-адреси та геолокації, часові характеристики, відомості про клієнтський пристрій, а також дані про попередні сесії, якщо вони зберігаються у локальній базі.

На основі зібраної інформації виконується обчислення показника довіри згідно з математичною моделлю, що буде докладно розглянута в наступному розділі. Якщо одержане значення перевищує встановлений верхній поріг, шлюз надає клієнту повний доступ до файлового сервісу, наприклад шляхом видачі тимчасового токена або формування спеціального посилання, дійсного протягом обмеженого часу. Якщо показник довіри потрапляє в проміжний інтервал, система може запропонувати додаткові кроки перевірки (двофакторну автентифікацію, підтвердження через довірений пристрій тощо) або надати обмежений набір функцій сервісу. У разі, коли trust-score менший за нижній поріг, доступ блокується, а у журналі фіксується відповідний інцидент.

Усі ключові дії під час обробки запиту - значення контекстних факторів, розрахований показник довіри, обраний рівень доступу та ідентифікатор клієнта - заносяться до журналу подій. Це дозволяє аналізувати поведінку системи, оцінювати коректність обраних порогів і ваг факторів, а також у перспективі використовувати накопичені дані для побудови більш складних моделей, зокрема на основі методів машинного навчання [27]. Таким чином, взаємодія компонентів архітектури забезпечує реалізацію динамічного, контекстно-залежного контролю доступу, що відповідає принципам Нульової довіри і адаптована до умов HomeLab-інфраструктури.

4. МАТЕМАТИЧНА МОДЕЛЬ ДИНАМІЧНОЇ ДОВІРИ

Математична модель динамічної довіри покликана формалізувати процес прийняття рішень щодо доступу до ресурсів HomeLab-інфраструктури на основі множини контекстних параметрів, що описують властивості користувача, пристрою та середовища доступу. Нехай кожній сесії доступу відповідає вектор нормалізованих контекстних ознак $\mathbf{f} = (f_1, f_2, \dots, f_n)^\top$, де $f_i \in [0; 1]$ - безрозмірні оцінки окремих факторів (геолокаційного, часовго, інтеграційного тощо), отримані в результаті перетворення сирих даних (IP-адреси, часової мітки, історії пристрою та ін.). Мета моделі полягає в тому, щоб побудувати відображення

$$F: [0; 1]^n \to [0; 1],$$

яке зіставляє кожному вектору $\mathbf{f}$ скалярне значення $T_{score} = F(\mathbf{f})$, що інтерпретується як інтегральний рівень довіри до поточної сесії доступу.

У цій роботі для F обрано лінійний агрегатор із невід'ємними ваговими коефіцієнтами, який зручно записати у векторній формі

$$T_{score} = \langle \mathbf{w}, \mathbf{f} \rangle = \mathbf{w}^\top \mathbf{f},$$

де $\mathbf{w} = (w_1, \dots, w_n)^\top$ - вектор ваг, такий що $w_i \geq 0$ і $\sum_{i=1}^{n} w_i = 1$. Така форма моделі забезпечує прозорість інтерпретації: внесок кожного фактора у підсумковий показник довіри прямо пропорційний відповідній вазі, а значення T_{score} завжди лежить в інтервалі $[0; 1]$ за рахунок нормування як факторів, так і ваг. Додатково вводиться функція ризику

$$R = 1 - T_{score},$$

яка інтерпретується як залишковий ризик, пов'язаний із дозволом поточної сесії доступу; чим більший T_{score}, тим менший R, і навпаки, що відповідає підходам risk-based access control у сучасних IAM-системах.

Рішення щодо рівня доступу приймається на основі порівняння T_{score} із парою порогів (T_{low}, T_{high}), що задають трирівневу градацію доступу. Формально можна визначити відображення

$$D: [0; 1] \to \{\text{Full,Limited,Blocked}\},$$

де

$$D(T_{score}) = \begin{cases} \text{Full}, & T_{score} \in [T_{high}, 1], \\ \text{Limited}, & T_{score} \in [T_{low}, T_{high}), \\ \text{Blocked}, & T_{score} \in [0, T_{low}). \end{cases}$$

Таке багаторівневе рішення узгоджується з практикою Zero Trust Network Access, де для низькоризикових сесій надається розширений функціонал, для проміжних - обмежений, а високоризикові спроби блокуються або вимагають додаткових перевірок [10]. У реалізації шлюзу FastAPI значення T_{score} та відповідне $D(T_{score})$ зберігаються в таблиці AccessEvent для кожної події доступу, що дозволяє проводити подальший аналіз та налаштування моделі.

4.1 Контекстні фактори

Запропонована система динамічного контролю доступу ґрунтується на багатовимірному описі контексту сеансу підключення. Для кожного запиту мобільного клієнта формується вектор контекстних факторів, значення яких надалі використовуються для обчислення інтегрального показника довіри.

На основі аналізу літератури та вимог до HomeLab-інфраструктури виділено три базові групи факторів, які можна отримати без додаткового спеціалізованого обладнання:

- **геолокаційний контекст** - інформація про приблизне розташування клієнта, що визначається на основі IP-адреси або супутніх даних;
- **часовий контекст** - час доби, день тижня та інші часові ознаки, що дозволяють моделювати «типову» активність користувача;
- **контекст цілісності пристрою** - узагальнена оцінка стану клієнтського пристрою (наявність блокування екрана, використання актуальної версії операційної системи, відсутність ознак компрометації тощо).

Кожен фактор перетворюється на нормалізовану безрозмірну величину в інтервалі від 0 до 1, де 0 відповідає повній недовірі щодо відповідного аспекту контексту, а 1 - максимально можливій довірі. Наприклад, геолокаційний фактор може набувати

значення 1 для «домашньої» мережі, проміжних значень для відомих безпечних локацій і наближатися до 0 для невідомих або потенційно ризикованих регіонів. Часовий фактор будується на основі історії взаємодії й відображає, наскільки поточний час підключення відповідає типовим для користувача інтервалам активності. Фактор цілісності пристрою формується з урахуванням кількох бінарних чи градуйованих ознак (наявність блокування екрана, стан оновлень, наявність root/jailbreak та ін.) і також приводиться до відрізка [0;1], і модель прорахунку доступу можна зобразити так як на Рисунку 4.1

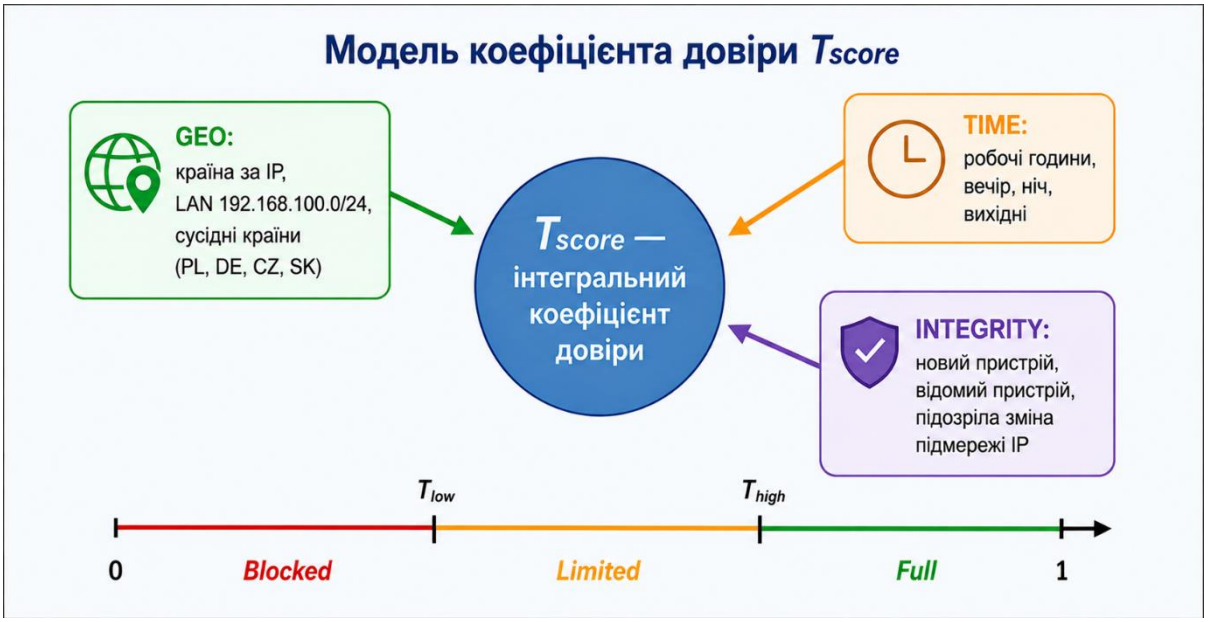

Рис. 4.1 Модель коефіцієнта довіри Tscore

4.2 Формула коефіцієнта довіри

Для HomeLab-середовища, де обсяг доступних даних для навчання складних моделей є обмеженим, доцільно застосовувати прозору і легко інтерпретовану лінійну модель інтегрального показника довіри. У загальному вигляді показник довіри T_{score} може бути подано як зважену суму нормалізованих факторів:

$$T_{score} = w_{geo} \cdot f_{geo} + w_{time} \cdot f_{time} + w_{int} \cdot f_{int},$$

де $f_{geo}, f_{time}, f_{int}$ - нормалізовані значення геолокаційного, часового та інтеграційного факторів відповідно, а $w_{geo}, w_{time}, w_{int}$ - їх вагові коефіцієнти, що задовольняють умову нормування $w_{geo} + w_{time} + w_{int} = 1$.

Вагові коефіцієнти обираються на основі експертного аналізу й відображають відносну важливість відповідних факторів для конкретної інфраструктури. Наприклад, для домашньої лабораторії, де критичною є захищеність пристрою користувача, але геолокація суттєво не змінюється, може бути доцільно надати більшої ваги фактору цілісності пристрою, дещо меншої - часовому фактору й найменшої - геолокаційному. Значення коефіцієнтів можуть уточнюватися в процесі експериментальної експлуатації системи на основі аналізу журналів подій.

На основі T_{score} вводяться три інтервали значень, що відповідають рівням доступу:

- $T_{score} \geq T_{\text{high}}$ - **високий рівень довіри**, надається повний доступ;
- $T_{\text{low}} \leq T_{score} < T_{\text{high}}$ –-**проміжний рівень**, надається обмежений доступ або ініціюються додаткові перевірки;
- $T_{score} < T_{\text{low}}$ - **низький рівень довіри**, доступ блокується.

Конкретні значення порогів T_{high} і T_{low} визначаються експериментально з урахуванням бажаного компромісу між рівнем безпеки та зручністю роботи користувачів. Аналіз журналів дозволяє виявити, наскільки часто легітимні користувачі потрапляють у «підозрілий» інтервал, і відповідно скоригувати порогові значення.

4.3 Логіка прийняття рішень

На основі обчисленого інтегрального показника довіри система повинна віднести поточний запит до одного з фіксованих рівнів доступу. У запропонованій моделі використовується триступенева шкала: високий рівень довіри (доступ до

чутливих ресурсів), проміжний рівень (звичайний доступ) та низький рівень (відмова в доступі до чутливого сегмента або повна блокада). Межі між цими рівнями визначаються двома порогами, що зберігаються в активному правилі довіри: верхній поріг відділяє зону високої довіри, а нижній - зону, де доступ до чутливих ресурсів заборонений.

У реалізації шлюзу значення інтегрального показника trust_score порівнюється з порогами активного правила. Якщо trust_score перевищує верхній поріг, запит класифікується як такий, що заслуговує на підвищений рівень довіри, й шлюз може дозволити доступ до розширеного набору функцій файлового сервісу. Якщо показник нижчий за нижній поріг, запит вважається ризиковим, і доступ до чутливих даних забороняється, а в окремих випадках може бути заблоковано будь-яку взаємодію з ресурсом. Для проміжних значень використовується нормальний рівень доступу, який забезпечує базову функціональність сервісу без підвищених привілеїв.

Зазначена логіка вбудована в основний маршрут шлюзу, що обробляє запити доступу. Після надходження HTTP-запиту шлюз спочатку визначає клієнтську IP-адресу, знаходить або створює записи про користувача й пристрій у локальній базі даних та обчислює часткові оцінки геолокаційного, часовго й інтеграційного факторів. Далі застосовується механізм обмеження частоти спроб входу: у разі надмірної кількості спроб за короткий час запит автоматично потрапляє в категорію з нульовим рівнем довіри, що запобігає атакам типу brute-force. Для запитів, які не підпадають під обмеження частоти, здійснюється перевірка існування користувача у зовнішній системі зберігання, після чого на основі trust_score і порогів формується остаточне рішення щодо рівня доступу.
Крім власне рішення («дозволити з підвищеними правами», «дозволити на базовому рівні», «відмовити»), шлюз формує текстове пояснення причини, в якому окремо зазначаються впливові фактори: нетипова країна, нічний або вихідний час, новий пристрій, зміна підмережі тощо.

5. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОТОТИПУ

У попередніх розділах було сформульовано вимоги до системи динамічного контролю доступу в середовищі HomeLab, запропоновано архітектуру та розроблено математичну модель оцінки довіри мобільного клієнта. На цьому етапі робота переходить від концептуального рівня до практичної реалізації: створення повноцінного прототипу, його розгортання у реальному лабораторному середовищі та проведення експериментальних досліджень.

Метою цього розділу є опис процесу реалізації запропонованої системи, характеристика використаних технічних засобів та програмного забезпечення, а також представлення результатів експериментів, які підтверджують працездатність моделі динамічної довіри та її відповідність поставленим вимогам. Особлива увага приділяється інтеграції логічного шлюзу довіри з приватним хмарним файловим сервісом і зовнішньою Zero Trust-інфраструктурою, а також аналізу поведінки системи в різних сценаріях підключення.

У подальших підрозділах буде послідовно розглянуто апаратно-програмну платформу HomeLab-стенда, структуру та ключові компоненти реалізованого прототипу, методику проведення експериментів і отримані результати. Такий підхід дозволяє простежити шлях від теоретичної моделі до її практичного втілення та дати обґрунтовану оцінку ефективності запропонованого рішення.

5.1 Технічні засоби й ПЗ

Реалізація прототипу виконувалася на базі існуючої HomeLab-інфраструктури, до складу якої входить сервер віртуалізації з гіпервізором, маршрутизатор, керований комутатор та локальний сегмент мережі з адресним простором 192.168.100.0/24. На фізичному сервері розгорнуто кілька віртуальних машин, серед яких виділено окремі вузли для приватного хмарного файлового сервісу та шлюзу динамічної довіри. Така конфігурація забезпечує

логічну та мережеву ізоляцію компонентів, а також дозволяє гнучко змінювати їх параметри без впливу на інші служби HomeLab.

У ролі цільового ресурсу використано самохостинговий хмарний файловий сервіс, розгорнутий на окремій віртуальній машині у внутрішньому сегменті (зокрема, на адресі виду 192.168.100.248). Цей сервіс надає веб-інтерфейс і клієнтські застосунки, підтримує користувацькі облікові записи та групи, а також має API для керування користувачами й групами доступу. Саме можливість програмного керування обліковими записами й групами через API була одним із критеріїв вибору, оскільки вона дозволяє шлюзу динамічної довіри синхронізувати рішення про рівень довіри з фактичними правами доступу до сховища.

Логічний шлюз динамічної довіри реалізовано як веб-службу на окремому віртуальному або фізичному вузлі, доступному з внутрішнього сегмента мережі під доменним іменем чи IP-адресою виду 192.168.100.x. Для розроблення бекенд-частини використано мову програмування Python 3 та фреймворк FastAPI, який забезпечує декларативне описання REST-інтерфейсів, асинхронну обробку запитів, інтеграцію з Pydantic для валідації вхідних і вихідних структур даних, а також зручну взаємодію з ORM-шаром на базі SQLAlchemy [18-19,29]. Локальна база даних шлюзу містить таблиці користувачів, пристроїв, правил довіри та журналів подій, що дозволяє зберігати історію сесій і параметри контексту для подальшого аналізу.

Для визначення геолокації за IP-адресою використано бібліотеку GeoIP2 з локальною базою GeoLite2-City, а у випадку її відсутності застосовується спрощений евристичний підхід, який класифікує приватні адресні простори як локальну мережу [20,26]. Часовий контекст базується на системному часові сервера й класифікації моменту підключення за добовими й тижневими інтервалами, тоді як контекст цілісності пристрою формується на основі записів про першу та останню появу пристрою в системі й аналізу зміни підмережі його IP-адреси.

Зовнішній доступ до HomeLab-інфраструктури організовано через сервіси Zero Trust-класу, зокрема за допомогою тунелювання HTTP(S)-трафіку від внутрішнього шлюзу до хмарної інфраструктури провайдера без відкриття вхідних

портів на домашньому маршрутизаторі [25]. На рівні цих сервісів додатково можуть застосовуватися політики аутентифікації користувачів (наприклад, через зовнішній Identity-провайдер), що утворює додатковий бар’єр безпеки перед тим, як запит потрапляє до шлюзу динамічної довіри. Такий підхід поєднує переваги промислових Zero Trust-рішень із гнучкістю і прозорістю власної моделі довіри, реалізованої в межах HomeLab.

Таким чином, обрана сукупність апаратних та програмних засобів забезпечує повноцінне середовище для реалізації й дослідження запропонованої моделі динамічної довіри: гіпервізор бере на себе функції ізоляції та гнучкого керування ресурсами, приватний хмарний сервіс виступає критичним ресурсом зберігання даних, шлюз на базі FastAPI реалізує математичну модель та логіку прийняття рішень, а Zero Trust-тунель надає безпечний зовнішній доступ до цієї інфраструктури.

5.2 Реалізація компонентів та Сценарії тестування

Архітектура реалізованого прототипу відображає трирівневу модель, описану в розділі 3: клієнтський рівень, рівень шлюзу динамічної довіри та рівень сервісу зберігання даних, обгорнутий Zero Trust-інфраструктурою. Усі компоненти розгорнуті в межах HomeLab-стенда на базі гіпервізора, що забезпечує окремі віртуальні машини для приватного хмарного сховища та шлюзу, тоді як зовнішній доступ організовано через тунель до хмарного провайдера без відкриття вхідних портів на домашньому маршрутизаторі.

Приватний файловий сервіс працює у внутрішньому сегменті мережі та надає як веб-інтерфейс, так і програмний інтерфейс керування користувачами й групами. У межах прототипу використано можливості Provisioning API для перевірки існування облікових записів і керування групою підвищеної довіри, що дає змогу синхронізувати рішення шлюзу (наприклад, «високий рівень довіри») із

фактичними правами доступу користувача в системі зберігання даних [23]. Завдяки цьому мережеве сховище залишається «простим» сервером файлів, а вся «інтелектуальна» логіка контролю доступу реалізується поза ним.

Шлюз динамічної довіри реалізовано як окремий веб-сервіс на основі фреймворку FastAPI з набором REST-маршрутів для перевірки доступу, діагностики та видачі HTML-порталу. Центральний маршрут /api/access/check приймає запит від клієнтського інтерфейсу із зазначенням логіна, відбитка пристрою та, за потреби, IP-адреси; доповнює його фактичним IP з контексту HTTP-з'єднання; знаходить або створює відповідні записи користувача й пристрою в локальній базі даних; обчислює часткові оцінки геолокаційного, часового та інтеграційного факторів; формує інтегральний показник довіри й на його основі ухвалює рішення щодо рівня доступу. Додатково в цьому ж маршруті реалізовано захист від надмірної кількості спроб входу за короткий інтервал часу, що дозволяє ранньо відсікати підозрілу активність типу brute-force незалежно від значення контекстних факторів.

Усі параметри сесії - IP-адреса, країна, часовий кошик, статус цілісності пристрою, значення часткових факторів, інтегральний показник довіри, прийняте рішення й сформоване текстове пояснення - зберігаються в таблиці журналу подій AccessEvent локальної бази даних. Це забезпечує повну трасованість рішень системи, а також створює підґрунтя для подальшого аналізу й уточнення вагових коефіцієнтів та порогів моделі. Окремий HTML-портал шлюзу використовується як «людський» інтерфейс, через який користувач може ініціювати процес входу, отримати відображення рішення системи й, за потреби, пояснення, чому певний доступ було обмежено.

Зовнішній рівень архітектури представлений Zero Trust-тунелем, який встановлює захищений канал між домашньою інфраструктурою та точкою присутності хмарного провайдера. На стороні провайдера налаштовано публічне доменне ім'я, яке резолюється на тунель, а на стороні HomeLab - агент тунелю, що приймає та переспрямовує HTTP(S)-трафік на шлюз динамічної довіри. За потреби на цьому рівні можуть додатково застосовуватися політики автентифікації

(наприклад, вимога входу через сторонній Identity-провайдер), однак у рамках прототипу основний акцент зроблено саме на контекстно-орієнтованій моделі довіри, реалізованій у власному шлюзі. У підсумку отримано інтегровану систему, в якій промислова Zero Trust-інфраструктура відповідає за безпечний мережевий доступ, а власний шлюз динамічної довіри - за контекстно-залежне керування рівнем доступу до приватного хмарного сховища

5.3 Аналіз результатів моделювання

Для оцінювання працездатності та адекватності запропонованої моделі динамічної довіри було сформовано набір типових сценаріїв тестування, що відображають як нормальну роботу системи, так і потенційно небажану або аномальну активність. Такий підхід відповідає загальним рекомендаціям щодо валідації моделей trust-scoring та ризик-орієнтованої автентифікації, де важливо перевірити поведінку системи в різних частинах простору контекстних параметрів.

Усі експерименти проводилися на реальному HomeLab-стенді з використанням фактичної мережевої топології, описаної в підрозділі 5.1. Для кожного сценарію фіксувалися вхідні параметри обчислені часткові оцінки факторів, інтегральний показник довіри trust_score, прийняте рішення про рівень доступу та сформоване текстове пояснення, які зберігалися в таблиці журналу подій AccessEvent. Це дозволило не лише перевірити коректність роботи алгоритмів, але й накопичити дані для подальшого аналізу чутливості моделі до змін окремих параметрів. (Рис 5.1)

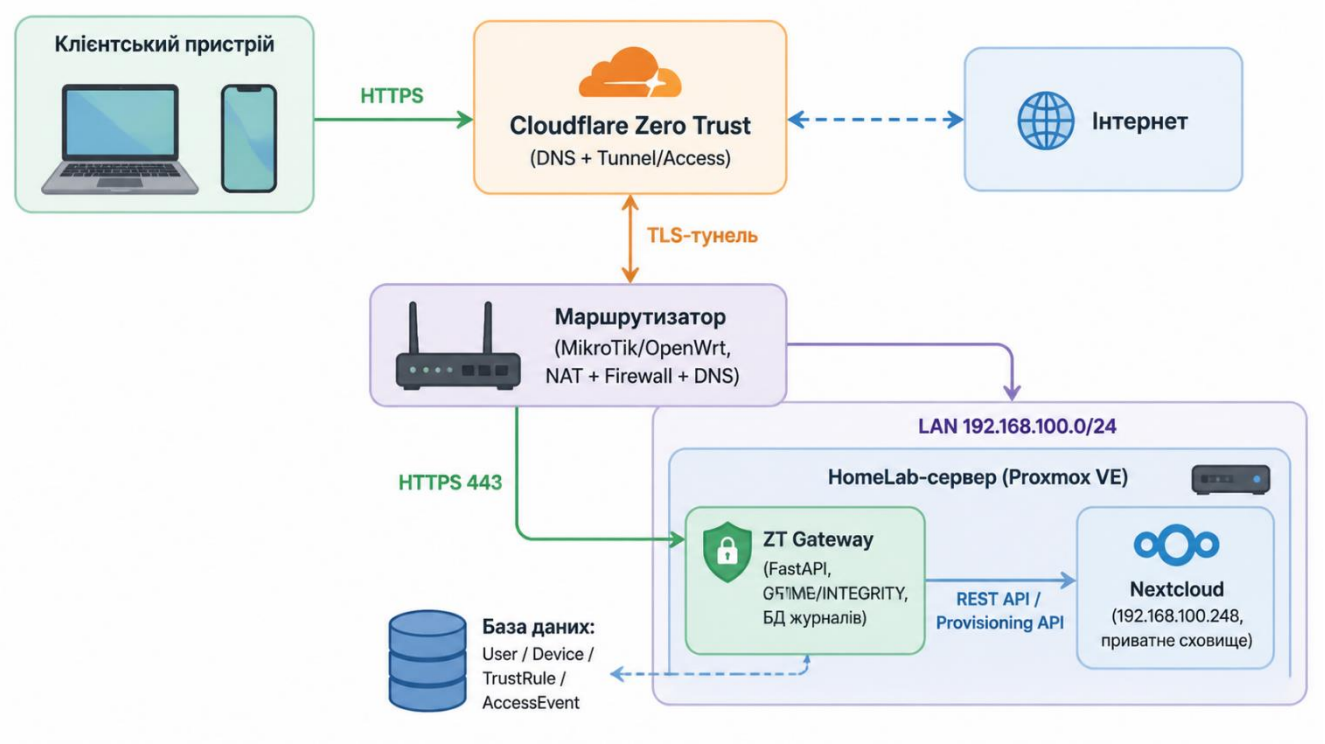

Рис. 5.1. Інфографіка з архітектурою Zero-Trust HomeLab з Cloudflare, маршрутизатором, Proxmox, ZT Gateway та Nextcloud

До основних сценаріїв тестування належать:

1. **Нормальний доступ з довіреного середовища.** У цьому сценарії користувач входить із локальної мережі домашнього сегмента в робочий час або ввечері, використовуючи пристрій, який уже неодноразово з'являвся в системі й не змінював підмережу IP-адреси. Очікується, що геолокаційний, часовий та інтеграційний фактори набувають високих значень, інтегральний показник довіри перевищує верхній поріг, а система надає підвищений рівень доступу до чутливих ресурсів. Цей сценарій використовується як «еталонний» для перевірки того, що легітимні користувачі в типових умовах не зазнають зайвих обмежень. Це зображено на Рисунку 5.2

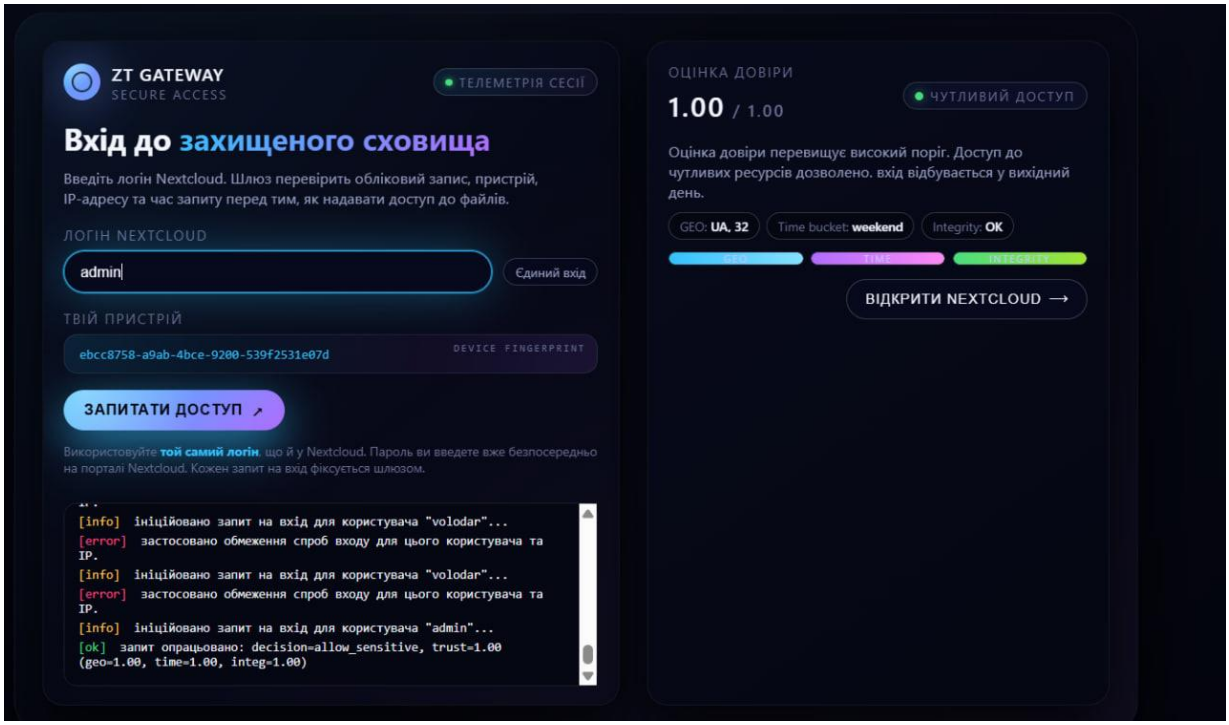

Рис. 5.2 – повний доступ

2. **Новий пристрій у звичній мережі.** У цьому випадку вхід здійснюється з локальної мережі, але з нового відбитка пристрою, який щойно з’явився в системі. Згідно з реалізованою моделлю, статус цілісності такого пристрою позначається як новий, а відповідний фактор отримує занижену оцінку, тоді як геолокаційний і часовий фактори можуть залишатися високими. Очікується, що інтегральний показник довіри потрапляє в проміжний діапазон між нижнім і верхнім порогами, система надає звичайний або обмежений рівень доступу та формує пояснення, у якому окремо зазначено використання нового пристрою. Такий сценарій демонструє здатність системи обережно поводитися з новими пристроями, не блокуючи одразу легітимних користувачів, Рисунок 5.3

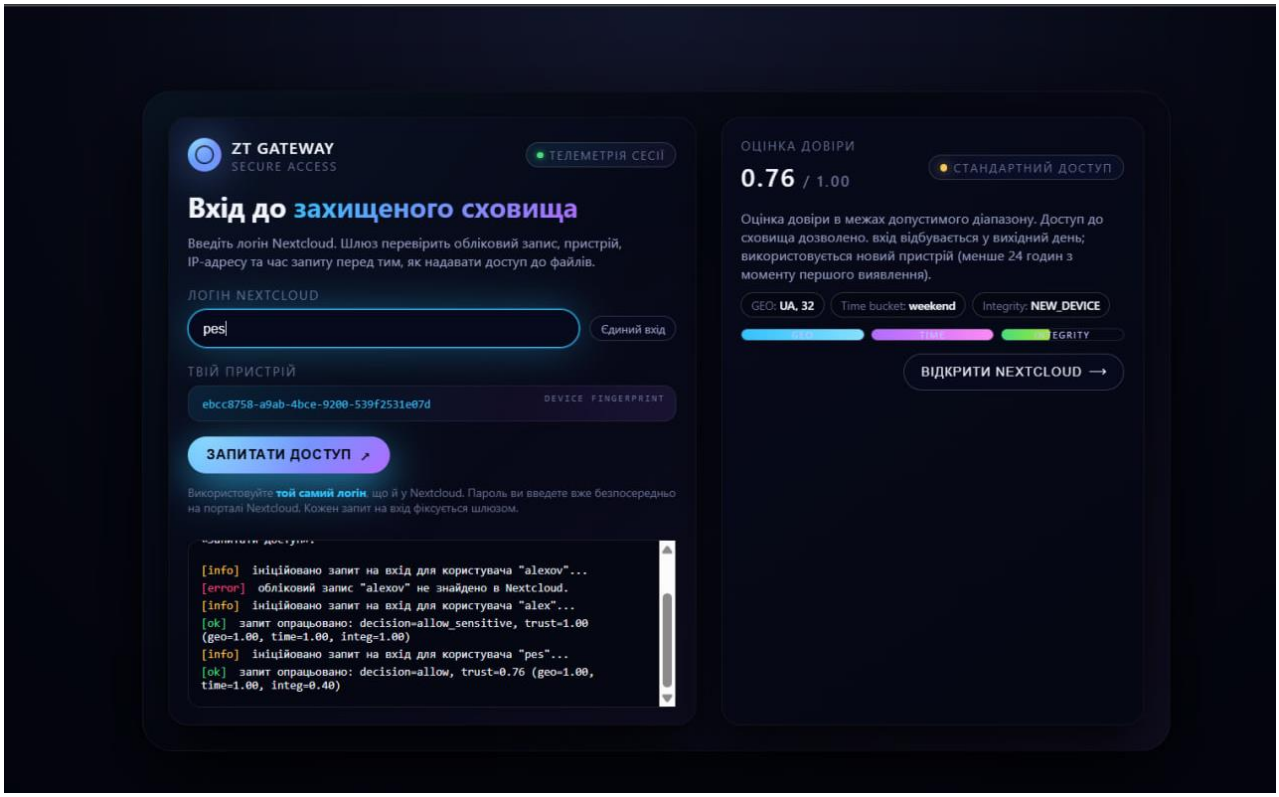

Рис. 5.3 – логін з нового пристрою

3. **Доступ з нетипового географічного регіону.** У цьому сценарії моделюється ситуація, коли користувач намагається ввійти до системи з IP-адреси, що належить до країни або регіону, які раніше не спостерігалися у журналах подій. Для таких випадків геолокаційний фактор отримує знижений бал, тоді як часовий та інтеграційний фактори можуть залишатися в межах норми. Очікується, що інтегральний показник довіри зменшиться, а рішення системи перемкнеться на звичайний або навіть відмовний рівень доступу залежно від обраних ваг і порогів. У поясненні до події повинна відображатись інформація про нетипову країну доступу.
4. **Зміна мережевої підмережі для відомого пристрою.** Цей сценарій перевіряє реакцію системи на зміну мережевого оточення. Вхід виконується з пристрою, який уже неодноразово використовувався, але тепер знаходиться в іншій підмережі (наприклад, перехід з домашньої Wi-Fi-мережі на гостьову або мобільну мережу). У моделі це відображається як статус цілісності «підозріла зміна IP-підмережі» з проміжною оцінкою, що зменшує інтегральний показник довіри порівняно з базовим сценарієм. У

журналі подій має бути зафіксовано як зміну підмережі, так і вплив цього фактора на рішення системи.

5. **Сценарії з високою частотою спроб (rate limiting).** Окрему групу тестів становлять сценарії, у яких за короткий час здійснюється низка послідовних спроб входу з однаковими або різними обліковими записами, але з однієї IP-адреси. Відповідно до реалізації шлюзу, якщо кількість спроб за фіксований інтервал перевищує встановлений поріг, система примусово встановлює показник довіри в нуль і класифікує подію як обмежену за частотою, не звертаючись до зовнішнього сервісу для перевірки облікового запису[30]. Такий механізм дозволяє ефективно захищатися від атак перебору паролів незалежно від конкретних значень контекстних факторів.
6. **Невідомий або неіснуючий обліковий запис.** У цьому сценарії перевіряється ситуація, коли клієнт надсилає запит з логіном, що відсутній у зовнішньому файловому сервісі. Навіть якщо контекстні фактори формально виглядають прийнятними, перевірка через API зовнішнього сервісу призводить до відмови в довірі, а інтегральний показник довіри примусово встановлюється в нуль. Це відповідає політиці «fail-closed», коли в разі невизначеності або невідповідності ідентифікаційних даних система не надає доступу.

Для кожного з наведених сценаріїв результати фіксувалися у вигляді записів у базі даних та, за потреби, додатково експортувалися для подальшого аналізу. Порівняння фактичних рішень системи з очікуваною поведінкою дозволило підтвердити, що розроблена модель динамічної довіри адекватно реагує на зміни контексту, підвищуючи або знижуючи рівень доступу залежно від комбінації факторів, а не лише від факту пред’явлення коректних облікових даних [2,16]. Це, у свою чергу, створює підґрунтя для подальшого розширення моделі за рахунок додаткових факторів та використання накопичених журналів для тонкого налаштування ваг і порогів.

ВИСНОВОК

У роботі розроблено модель динамічної довіри для HomeLab, що реалізує принципи Нульової довіри (Zero Trust) через контекстно-адаптивну верифікацію клієнтів. Запропонований підхід ґрунтується на обчисленні інтегрального показника T_{score}, який узагальнює геолокаційні, часові та інтеграційні фактори для прийняття рішень про рівень доступу до Nextcloud-сервісу. Модель забезпечує прозору інтерпретацію рішень, що дозволяє адміністратору гнучко налаштовувати безпеку без використання складних комерційних систем.

Практична реалізація виконана у вигляді програмного шлюзу на FastAPI, інтегрованого з Nextcloud через Provisioning API та бази даних подій. Система підтримує політику fail-closed, захист від brute-force атак (rate limiting) та динамічне керування правами користувачів. Експериментальне тестування на репрезентативних сценаріях підтвердило, що модель забезпечує контекстну гнучкість і безпеку, значно перевищуючи за ефективністю статичні списки контролю доступу (ACL).

Розроблені рішення є масштабованими для подальшого розширення новими контекстними факторами (Play Integrity API, поведінкові ознаки) та впровадження складніших методів оцінки ризику. Робота демонструє, що принципи Нульової довіри можуть бути успішно адаптовані для домашніх лабораторій, створюючи надійну основу для безпечного керування особистими даними та сервісами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Stallings W. Network Security Essentials: Applications and Standards. 6th ed. Pearson, 2017. 448 p. Дата звернення: 09.04.2026

2. Rose S., Borchert O., Mitchell S., Connelly S. Zero Trust Architecture. NIST Special Publication 800-207. National Institute of Standards and Technology. 2020. Дата звернення: 09.04.2026 Джерело: https://csrc.nist.gov/pubs/sp/800/207/final

3. Tanenbaum A. S., Wetherall D. Computer Networks. 5th ed. Prentice Hall, 2011. 933 p. Дата звернення: 09.04.2026

4. Kindervag J. No More Chewy Centers: Introducing The Zero Trust Model Of Information Security. Forrester Research. 2010. Дата звернення: 09.04.2026 Джерело: https://media.paloaltonetworks.com/documents/Forrester-No-More-Chewy-Centers.pdf

5. Kindervag J. Build Security Into Your Network's DNA: The Zero Trust Network Architecture. Forrester Research, 2010. Дата звернення: 17.05.2026 Джерело: https://www.ndm.net/firewall/pdf/palo_alto/Forrester-Applying-Zero-Trust.pdf

6. BeyondCorp Zero Trust Enterprise Security. Google Cloud. Дата звернення: 17.05.2026. Джерело: https://cloud.google.com/beyondcorp

7. Azure Conditional Access Documentation. Microsoft. Дата звернення: 17.05.2026. Джерело: https://learn.microsoft.com/en-us/entra/identity/conditional-access/overview

8. Understanding Adaptive Authentication and How It Works. Okta. Дата звернення: 17.05.2026. Джерело: https://www.okta.com/identity-101/adaptive-authentication/

9. Zero Trust Reference Architecture. Cloudflare Docs. Дата звернення: 17.05.2026. Джерело: https://developers.cloudflare.com/reference-architecture/implementation-guides/zero-trust/

10. Grassi P. A. et al. Digital Identity Guidelines. NIST Special Publication 800-63-4. National Institute of Standards and Technology, 2025. Дата звернення: 17.05.2026 Джерело: https://csrc.nist.gov/pubs/sp/800/63/4/final

11. Okta launches risk-based authentication solution. Datacentre Solutions. Дата звернення: 17.05.2026. Джерело: https://datacentre.solutions/news/56643/okta-launches-risk-based-authentication-solution

12. FAR vs FRR: Biometric Error Rates Explained. Signzy. Дата звернення: 17.05.2026. Джерело: https://www.signzy.com/general-glossary/far-frr-biometric-error-rates

13. LUKS On-Disk Format Specification. cryptsetup / GitLab kernel.org. Дата звернення: 17.05.2026. Джерело: https://gitlab.com/cryptsetup/cryptsetup/-/wikis/LUKS-standard/on-disk-format.pdf

14. VeraCrypt Documentation. VeraCrypt. Дата звернення: 17.05.2026. Джерело: https://www.veracrypt.fr/en/Documentation.html

15. Proxmox VE Administration Guide. Proxmox Server Solutions GmbH. Дата звернення: 17.05.2026. Джерело: https://pve.proxmox.com/pve-docs/pve-admin-guide.html

16. Городицький О. Покроковий підхід до імплементації Zero Trust. Кібербезпека: освіта, наука, техніка. 2025. № 4(30). Джерело: https://csecurity.kubg.edu.ua/index.php/journal/article/view/1029

17. Okta. The State of Secure Identity 2023. Okta Inc., 2023. Дата звернення: 17.05.2026. Джерело: https://www.okta.com/au/resources/whitepaper-the-state-of-secure-identity-report/

18. FastAPI Documentation. Tiangolo. Дата звернення: 17.05.2026. Джерело: https://fastapi.tiangolo.com

19. SQLAlchemy — The Python SQL Toolkit and ORM. Дата звернення: 17.05.2026. Джерело: https://www.sqlalchemy.org

20. GeoLite2 Free Geolocation Data. MaxMind Developer Portal. Дата звернення: 17.05.2026. Джерело: https://dev.maxmind.com/geoip/geolite2-free-geolocation-data/

21. Play Integrity API. Google Developers. Дата звернення: 17.05.2026. Джерело: https://developer.android.com/google/play/integrity/overview

22. Firewall. MikroTik RouterOS Documentation. Дата звернення: 17.05.2026. Джерело: https://help.mikrotik.com/docs/spaces/ROS/pages/250708066/Firewall

23. User Provisioning API. Nextcloud Administration Manual. Дата звернення: 17.05.2026. Джерело: https://docs.nextcloud.com/server/31/admin_manual/configuration_user/user_provisioning_api.html

24. GeoIP and GeoLite City and Country Databases. MaxMind Developer Portal. Дата звернення: 17.05.2026. Джерело: https://dev.maxmind.com/geoip/docs/databases/city-and-country/

25. Cloudflare Tunnel Documentation. Cloudflare Docs. Дата звернення: 17.05.2026. Джерело: https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/

26. MaxMind GeoIP2 Python API — geoip2 Documentation. Дата звернення: 17.05.2026. Джерело: https://geoip2.readthedocs.io/en/latest/

27. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 800 с. Дата звернення Джерело: https://www.deeplearningbook.org

28. Chen Y., Paxson V., Katz R. What's New About Cloud Computing Security? Technical Report UCB/EECS-2010-5. UC Berkeley. 2010. Джерело: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-5.pdf

29. Pydantic V2 Documentation. Pydantic. Дата звернення: 17.05.2026. Джерело: https://docs.pydantic.dev/latest/

30. OWASP Authentication Cheat Sheet. OWASP Foundation. Дата звернення: 17.05.2026. Джерело: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

ДОДАТОК

Додаток А. Вибрані фрагменти коду

```
def compute_geo_score(geo_country: str, ip: str) -> float:
    if ip.startswith("192.168.100."):
        return 1.0
    if geo_country in ("UA", "LAN"):
        return 1.0
    if geo_country in ("PL", "DE", "CZ", "SK"):
        return 0.7
    if geo_country == "UNKNOWN":
        return 0.5
    return 0.3
```

Пояснення: Функція перетворює країну та тип мережі на нормований бал f_geo ∈ [0; 1]. Локальна підмережа та Україна отримують максимальну оцінку 1.0, сусідні довірені країни - 0.7, невідома геолокація - 0.5, решта - 0.3. Це ілюструє принцип контекстно-орієнтованої безпеки.

```
def classify_time_bucket_and_score(ts: datetime) -> Tuple[str, float]:
    weekday = ts.weekday()  # 0=Monday ... 6=Sunday
    hour = ts.hour
    if weekday >= 5:
        return "weekend", 1.0
    if 8 <= hour < 20:
        return "work_hours", 1.0
    if 20 <= hour < 23:
        return "evening", 0.7
    return "night", 0.3
```

Пояснення: Функція класифікує поточний момент у часовий інтервал і повертає пару (bucket, f_time). Вихідні та робочі години вважаються типовими (1.0), вечір знижує бал до 0.7, нічний час - до 0.3. Назва інтервалу зберігається у журналі подій AccessEvent для ретроспективного аналізу.

```
def compute_integrity_status_and_score(
    device: models.Device, current_ip: str, now: datetime
) -> Tuple[str, float]:
    age_seconds = (now - device.first_seen).total_seconds()
    if age_seconds < 24 * 3600:
        return "NEW_DEVICE", 0.4
    if _ip_subnet24(device.last_ip) == _ip_subnet24(current_ip):
        return "OK", 1.0
    return "SUSPICIOUS_IP_CHANGE", 0.6
```

Пояснення: Функція аналізує відомість пристрою через fingerprint та зміну підмережі /24. Новий пристрій (менше 24 годин з першої появи) отримує знижений бал 0.4, відомий пристрій без зміни підмережі - 1.0, зміна підмережі - 0.6 як підозрілий сигнал. Цей компонент реалізує інтеграційний фактор f_int математичної моделі.

```
def compute_trust_score(rule, geo_score, time_score, integrity_score) -> float:
    return (
        rule.geo_weight * geo_score
        + rule.time_weight * time_score
        + rule.integrity_weight * integrity_score
    )

def decide_access(rule, score) -> Tuple[str, str, str]:
    if score >= rule.high_threshold:
        return "allow_sensitive", "sensitive", "trust_score >= high_threshold"
    if score < rule.low_threshold:
        return "deny_sensitive", "normal", "trust_score < low_threshold"
    return "allow", "normal", "low_threshold <= trust_score < high_threshold"
```

Пояснення: Перша функція безпосередньо реалізує формулу $T_{score} = w_{geo} \cdot f_{geo} + w_{time} \cdot f_{time} + w_{int} \cdot f_{int}$ з вагами, що зберігаються в таблиці TrustRule бази даних. Друга застосовує двопорогову логіку: high_threshold (за замовчуванням 0.85) відкриває чутливі ресурси, нижче low_threshold (0.5) - блокує.

```
recent_count = (
    db.query(models.AccessEvent)
    .filter(
        models.AccessEvent.user_id == user.id,
        models.AccessEvent.ip == client_ip,
        models.AccessEvent.login_time >= now - timedelta(seconds=10),
    )
    .count()
)

if recent_count >= 5:
    decision = "rate_limited"
    access_level = "none"
    trust_score = 0.0
else:
    user_exists_in_nc = nc_user_exists(req.username)
    if not user_exists_in_nc:
        decision = "invalid_username"
        trust_score = 0.0
    else:
        decision, access_level, _ = decide_access(rule, trust_score)
```

Пояснення: Перед будь-якими зверненнями до Nextcloud шлюз перевіряє частоту запитів — максимум 5 спроб за 10 секунд з одного user_id + IP. При перевищенні одразу виставляється trust_score = 0.0 без жодних API-викликів. Це реалізує принцип fail-closed: недоступність або помилка зовнішнього сервісу також призводить до відмови у доступі, а не до його надання за замовчуванням.