Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп’ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра

на тему: «Проєкт розподіленої корпоративної IT-інфраструктури з централізованою системою аутентифікації та управління доступом»

Виконав студент 4-го курсу групи КІ-22-1
спеціальності 123 «Комп'ютерна інженерія»
_____ Дмитро ГРИЩЕНКО
Керівник: Старший викладач кафедри КІ та КБ
_____ Марія КОЛОЩУК
Рецензент: Професор кафедри КІ та КБ, д.т.н., доцент
_____ Володимир ВОРОТНІКОВ

Житомир – 2026

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач
кафедри комп'ютерної
інженерії та кібербезпеки
_____ Андрій Єфіменко
2 лютого 2026 р.

ЗАВДАННЯ

на випускну кваліфікаційну роботу бакалавра

Студент: Грищенко Дмитро Сергійович
Тема роботи: «Проєкт розподіленої корпоративної IT-інфраструктури з централізованою системою аутентифікації та управління доступом»
затверджена Наказом університету від 2 лютого 2026 р. №57/с
Термін здачі студентом закінченої роботи 23 червня 2026 р.

Вихідні дані роботи:

- корпоративна мережа організації з розподіленою інфраструктурою та віддаленими користувачами;
- вимоги до системи захищеного віддаленого доступу на основі архітектури Zero Trust;
- сучасні кіберзагрози корпоративним мережам та вектори атак на VPN-рішення;
- стек відкритого програмного забезпечення: протокол WireGuard, VPN-портал Firezone, IAM-платформа Keycloak, міжмережевий екран pfSense;
- стандарти та протоколи: OpenID Connect (OIDC), OAuth 2.0, RBAC, PKCE;
- середовище моделювання — мережевий симулятор GNS3 з віртуальними машинами VirtualBox.

Обладнання та програмне забезпечення:

1. Маршрутизатор Cisco 7200 Series (граничні маршрутизатори R1, R2)
2. Міжмережевий екран pfSense CE 2.7
3. VPN-портал Firezone 0.7.x (WireGuard)
4. IAM-платформа Keycloak 24 (OIDC)
5. СУБД PostgreSQL 16
6. Середовище контейнеризації Docker + Docker Compose
7. Мережевий симулятор GNS3 2.2.x
8. Гіпервізор Oracle VirtualBox 7.0+
9. ОС вузлів Ubuntu Server 22.04 LTS; клієнтські ОС Windows 10, Ubuntu

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Єфіменко А.А.	4.03.2026	4.03.2026
2	Єфіменко А.А.	4.03.2026	4.03.2026
3	Єфіменко А.А.	4.03.2026	4.03.2026

Студент _____ Дмитро ГРИЩЕНКО

Керівник _____ Марія КОЛОЩУК

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз отриманого завдання та уточнення поставлених завдань	4.03.2026	Виконано
2	Аналіз літературних джерел	24.03.2026	Виконано
3	Розробка структурної моделі	02.04.2026	Виконано
4	Розрахунок параметрів адресації	04.04.2026	Виконано
5	Вибір технологій для реалізації	06.04.2026	Виконано
6	Проєктування КМ	21.04.2026	Виконано
7	Підготовка компонентів для мережі	04.05.2026	Виконано
8	Налаштування мережі	13.05.2026	Виконано
9	Налагодження захисту мережі	18.05.2026	Виконано
10	Тестування роботи КМ	26.05.2026	Виконано
11	Оформлення пояснювальної записки	27.05.2026	Виконано
12	Оформлення презентації	08.06.2026	Виконано
13	Захист	10-16.06.2026	Виконано

Студент _____ Дмитро ГРИЩЕНКО

Керівник _____ Марія КОЛОЩУК

АНОТАЦІЯ

Кваліфікаційна робота бакалавра містить відомості про систему, розроблену у межах даної роботи. Обсяг пояснювальної записки становить 101 сторінок, включає 51 рисунків та 19 таблиць. Список використаних джерел налічує 24 найменувань і розміщений на 2 сторінках.

Метою роботи є проєктування та практична реалізація системи захищеного віддаленого доступу до корпоративної мережі на основі архітектури Zero Trust із застосуванням WireGuard, Firezone та Keycloak у середовищі GNS3, а також верифікація її ефективності шляхом моделювання реальних векторів кібератак.

В результаті виконання роботи розгорнуто лабораторний стенд з чотирма ізольованими зонами безпеки, налаштовано pfSense CE 2.7 як засіб мережевої сегментації з правилами RBAC та Default Deny, реалізовано WireGuard-тунелювання через Firezone зі статичною прив'язкою тунельних адрес, розгорнуто Keycloak 24 як IAM-платформу з інтеграцією за протоколом OIDC Authorization Code Flow з PKCE. Тестування шести сценаріїв атак підтвердило ефективність системи та можливість реалізації принципів Zero Trust виключно засобами відкритого програмного забезпечення.

Результати роботи можуть бути використані для впровадження захищеного віддаленого доступу у корпоративних мережах підприємств малого та середнього бізнесу, а також як навчальна база для підготовки фахівців з мережевої безпеки.

Ключові слова: ZERO TRUST ARCHITECTURE, ЗАХИЩЕНИЙ ВІДДАЛЕНИЙ ДОСТУП, WIREGUARD, FIREZONE, KEYCLOAK, PFSENSE, OIDC, RBAC, МІКРОСЕГМЕНТАЦІЯ, GNS3, VPN-ІНФРАСТРУКТУРА, КІБЕРБЕЗПЕКА

ANNOTATION

The bachelor's qualification thesis presents a system developed within the scope of this work. The explanatory note comprises 101 pages, includes 51 figures and 19 tables. The list of references contains 24 entries and occupies 2 pages.

The goal of this work is to design and implement a secure remote access system for a corporate network based on Zero Trust Architecture using WireGuard, Firezone, and Keycloak in a GNS3 environment, and to verify its effectiveness by simulating real-world cyberattack vectors.

As a result, a laboratory testbed with four isolated security zones was deployed, pfSense CE 2.7 was configured as a network segmentation and traffic control component with RBAC and Default Deny policies, WireGuard tunneling was implemented via Firezone with static binding of tunnel addresses to user accounts, and Keycloak 24 was deployed as an IAM platform integrated via OIDC Authorization Code Flow with PKCE. Testing of six attack scenarios confirmed the system's effectiveness and the feasibility of implementing Zero Trust principles exclusively with open-source software.

The results can be applied to deploy secure remote access in small and medium-sized enterprise networks and serve as an educational foundation for training network security professionals.

Keywords: ZERO TRUST ARCHITECTURE, SECURE REMOTE ACCESS, WIREGUARD, FIREZONE, KEYCLOAK, PFSENSE, OIDC, RBAC, MICROSEGMENTATION, GNS3, VPN INFRASTRUCTURE, CYBERSECURITY

ЗМІСТ

ВСТУП 8

РОЗДІЛ 1. АНАЛІЗ ТЕХНОЛОГІЙ ЗАХИЩЕНОГО ВІДДАЛЕНОГО ДОСТУПУ ТА СИСТЕМ УПРАВЛІННЯ ІДЕНТИФІКАЦІЄЮ (IAM) 10

1.1 Аналіз сучасного стану загроз корпоративним мережам та обґрунтування актуальності теми 10

1.2 Підходи до організації захищеного віддаленого доступу 12

1.3 Порівняльний аналіз сучасних VPN-протоколів: WireGuard, OpenVPN та IPsec 17

1.4 Дослідження архітектури та можливостей сучасних IAM-систем на прикладі Keycloak 22

1.5 Вибір інструментальних засобів та формування вимог до апаратно-програмного комплексу 27

РОЗДІЛ 2. ПРОЄКТУВАННЯ КОРПОРАТИВНОЇ VPN-ІНФРАСТРУКТУРИ З ІНТЕГРАЦІЄЮ KEYCLOAK 31

2.1 Розробка загальної архітектури та топології захищеної мережі 31

2.2 Проєктування підсистеми віддаленого доступу на базі WireGuard та міжмережевого екрана pfSense 35

2.3 Розробка рольової моделі контролю доступу (RBAC) та варіанти використання системи 41

2.4 Алгоритми взаємодії компонентів мережі та проєктування централізованої автентифікації 45

2.5 Аналіз загроз для спроєктованої системи 52

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ ЗАХИЩЕНОГО ДОСТУПУ 58

3.1 Розгортання інфраструктури та базове налаштування мережевого шлюзу 58

3.2 Налаштування сервера автентифікації Keycloak та його інтеграція з VPN-сервісом 64

3.3 Налаштування політик журналювання подій та опис інтерфейсу керування доступом 72

3.4 Тестування системи: перевірка рольового доступу, аналіз стійкості та оцінка продуктивності 79

ВИСНОВКИ 93

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 95

ДОДАТКИ 98

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ACL (Access Control List) — список контролю доступу

BYOD (Bring Your Own Device) — концепція використання особистих пристроїв

DHCP (Dynamic Host Configuration Protocol) — протокол динамічного налаштування хосту

IAM (Identity and Access Management) — керування ідентифікацією та доступом

ICMP (Internet Control Message Protocol) — протокол керуючих повідомлень Інтернету

IKEv2 (Internet Key Exchange version 2) — протокол обміну ключами

JWT (JSON Web Token) — токен у форматі JSON

MTU (Maximum Transmission Unit) — максимальний розмір пакету

NAT (Network Address Translation) — трансляція мережевих адрес

OIDC (OpenID Connect) — протокол автентифікації на основі OAuth 2.0

PKCE (Proof Key for Code Exchange) — розширення безпеки OAuth 2.0

RBAC (Role-Based Access Control) — рольова модель керування доступом

SDP (Software Defined Perimeter) — програмно визначений периметр

SSO (Single Sign-On) — єдиний вхід

TLS (Transport Layer Security) — протокол захисту транспортного рівня

UDP (User Datagram Protocol) — протокол датаграм користувача

VPN (Virtual Private Network) — віртуальна приватна мережа

WAN (Wide Area Network) — глобальна мережа

ZTA / ZTNA (Zero Trust Architecture / Zero Trust Network Access) — архітектура нульової довіри

ВСТУП

Зростання кількості дистанційних робочих місць та перехід корпоративних сервісів до гібридних середовищ сформували критичну потребу у надійних механізмах захищеного віддаленого доступу. Згідно зі звітом Zscaler ThreatLabz 2025 VPN Risk Report, кількість задокументованих вразливостей (CVE) у VPN-рішеннях зросла на 82,5% за останній рік, причому близько 60% з них мають високий або критичний рівень серйозності, а 92% організацій фіксують атаки, у яких VPN-сервер слугував початковою точкою проникнення. Традиційна модель безпеки «замку та рову» (castle-and-moat) є неефективною проти сучасних загроз: компрометації облікових записів, атак на ланцюжок постачання та горизонтального переміщення зловмисника (lateral movement).

Концепція Zero Trust — «ніколи не довіряй, завжди перевіряй» — вимагає інтеграції трьох компонентів: захищеного транспортного каналу, централізованої IAM-системи та рольової фільтрації трафіку. Поєднання WireGuard з кодовою базою лише 4 000 рядків, open-source IAM-платформи Keycloak та міжмережевого екрана pfSense утворює стек, що технічно відповідає вимогам Zero Trust Network Access без залежності від хмарних SaaS-рішень. Дослідження та практична апробація такої архітектури є актуальним завданням для організацій, що прагнуть побудувати захищену інфраструктуру доступу на відкритому програмному забезпеченні.

Мета роботи — проєктування та практична реалізація системи захищеного віддаленого доступу до корпоративної мережі на основі архітектури Zero Trust із застосуванням протоколу WireGuard, сервісу Firezone та IAM-платформи Keycloak у середовищі мережевого симулятора GNS3, а також верифікація її ефективності шляхом моделювання реальних векторів кібератак.

Завдання роботи:

1. Виконати порівняльний аналіз VPN-протоколів (IPsec, OpenVPN, WireGuard) та концепцій захищеного доступу (VPN, ZTNA, SDP) для обґрунтування вибору технологічного стеку.

2. Проаналізувати можливості IAM-платформи Keycloak та її інтеграцію з VPN-сервісами через протокол OIDC.

3. Спроєктувати трирівневу архітектуру захищеного доступу: мережеву адресацію та матрицю рольового доступу, а також технічні механізми реалізації: ланцюжок NAT/Port Forward і правила фільтрації pfSense за принципом Default Deny.

4. Реалізувати інфраструктуру на стенді GNS3 з інтеграцією Firezone, Keycloak і pfSense, а також налаштуванням централізованої автентифікації через OIDC Authorization Code Flow з PKCE.

5. Провести комплексне тестування за шістьма сценаріями атак та сценаріями легітимного BYOD-доступу.

Об’єкт дослідження — процес забезпечення захищеного віддаленого доступу до корпоративних ресурсів в умовах розподіленої мережевої інфраструктури та розмитого периметра безпеки.

Предмет дослідження — методи та засоби побудови корпоративної VPN-інфраструктури на базі WireGuard, сервісу Firezone та IAM-системи Keycloak, що реалізують принципи Zero Trust і рольове розмежування доступу (RBAC) із централізованою автентифікацією за протоколом OIDC.

Методи дослідження — порівняльний аналіз протоколів та архітектурних концепцій; метод моделювання комп'ютерних мереж у середовищі GNS3; прецедентне проєктування (Use Case); тестування «чорної скриньки» шляхом відтворення реальних векторів атак; аналіз журналів подій для верифікації спрацювання правил безпеки.

Практичне значення одержаних результатів. Реалізована інфраструктура є відтворюваним зразком Zero Trust VPN-доступу на відкритому програмному забезпеченні. Розроблена архітектура може бути безпосередньо застосована в організаціях малого та середнього бізнесу, що потребують захищеного доступу для дистанційних співробітників без придбання комерційних ZTNA-рішень. Задокументовані конфігурації Cisco IOS ACL, правила pfSense та параметри інтеграції Keycloak–Firezone є практичними артефактами, придатними для розгортання в реальних корпоративних середовищах.

РОЗДІЛ 1. АНАЛІЗ ТЕХНОЛОГІЙ ЗАХИЩЕНОГО ВІДДАЛЕНОГО ДОСТУПУ ТА СИСТЕМ УПРАВЛІННЯ ІДЕНТИФІКАЦІЄЮ (IAM)

1.1 Аналіз сучасного стану загроз корпоративним мережам та обґрунтування актуальності теми

Стрімкий розвиток інформаційних технологій та масштабна цифровізація бізнес-процесів зумовили кардинальну трансформацію корпоративного середовища. Сучасні підприємства дедалі активніше впроваджують моделі гібридної та віддаленої роботи, залучають зовнішніх підрядників, використовують хмарні сервіси та розподілені інформаційні системи. Як наслідок, периметр корпоративної мережі перестав бути чітко визначеним, а традиційні підходи до захисту інформації — недостатніми. Паралельно з розширенням цифрової інфраструктури зростає кількість, різноманітність та складність кіберзагроз, спрямованих проти корпоративних мереж.

За даними Державної служби спеціального зв'язку та захисту інформації України, у 2024 році було зафіксовано 4315 кіберінцидентів — що на 70% більше порівняно з попереднім роком. У другій половині 2024 року кількість цілеспрямованих кібернападів на українські організації зросла на 48%, а масштаби автоматизованих атак збільшились із 567 атак на секунду у 2021 році до 7000 атак на секунду у 2024 році [1]. Глобальні витрати на забезпечення кібербезпеки досягли 212 мільярдів доларів у 2025 році, однак, попри значні інвестиції, кількість успішних атак продовжує зростати.

Серед найпоширеніших типів загроз корпоративним мережам виділяють: несанкціонований доступ через скомпрометовані облікові дані, атаки типу ransomware, фішинг, експлуатацію вразливостей у програмному забезпеченні для віддаленого доступу, а також атаки через ланцюжок постачання (supply chain attacks). Дослідження показують, що 48% усіх витоків даних у 2024 році стались через вразливості у механізмах доступу третіх сторін — підрядників та постачальників. При цьому 70% задокументованих інцидентів були пов'язані з

надмірно привілейованими обліковими записами, а 82% компаній неусвідомлено надавали стороннім користувачам доступ до всього хмарного середовища [2].

Окремої уваги заслуговує проблема безпеки рішень для захищеного віддаленого доступу — зокрема, традиційних VPN-систем. Незважаючи на широке застосування, класичні VPN-рішення демонструють суттєві вади в умовах сучасного ландшафту загроз. Згідно зі звітом Zscaler ThreatLabz 2025 VPN Risk Report, кількість задокументованих вразливостей (CVE) у VPN-рішеннях зросла на 82,5% за останній рік, причому близько 60% з них мають високий або критичний рівень серйозності за шкалою CVSS. При цьому 92% опитаних фахівців з кібербезпеки висловлюють занепокоєння тим, що незакриті вразливості VPN безпосередньо призводять до інцидентів із програмами-вимагачами, а 93% стурбовані ризиками backdoor-доступу через сторонні VPN-підключення [3]. Ключовою структурною проблемою є те, що традиційні VPN після успішної автентифікації надають користувачу широкий доступ до корпоративної мережі, що відкриває можливості для горизонтального переміщення (lateral movement) у разі компрометації облікових даних.

Проблема неналежної організації автентифікації та управління ідентифікацією є однією з центральних у сфері корпоративної безпеки. Дослідження свідчать, що 51% організацій не здійснюють належного контролю третіх сторін перед наданням їм доступу до систем, 35% не мають чіткого уявлення про те, яким чином відбуваються кібератаки, а 58% підприємств не мають послідовних стратегій управління привілейованим доступом [4]. Відсутність централізованої системи управління ідентифікацією та доступом (Identity and Access Management, IAM) унеможливлює ефективний контроль за тим, хто, коли і до яких ресурсів отримує доступ.

У відповідь на ці виклики галузева спільнота все більше звертається до концепції Zero Trust та технологій IAM. За даними HPE VPN Risk Report [4], 75% компаній розглядають Zero Trust як пріоритетний напрямок розвитку безпекової стратегії, а 81% організацій планують впровадити рішення на основі Zero Trust Network Access (ZTNA) до 2026 року [3]. Ключовим елементом реалізації таких

підходів є централізовані IAM-платформи, що забезпечують єдину точку управління автентифікацією, рольовим доступом та аудитом подій безпеки.

Отримані результати підтверджують актуальність і практичну значущість теми дослідження даної кваліфікаційної роботи. Проєктування корпоративної VPN-інфраструктури з централізованою автентифікацією на основі сучасних IAM-технологій дозволяє вирішити одразу кілька критичних проблем: усунути вразливості традиційних VPN-рішень, запровадити принцип найменших привілеїв через рольову модель доступу (RBAC), забезпечити повноцінне журналювання та аудит, а також наблизити архітектуру корпоративної мережі до сучасної парадигми Zero Trust. Практична реалізація такої системи на базі відкритого програмного забезпечення (WireGuard, Keycloak, pfSense) підтверджує доцільність та відтворюваність запропонованого підходу для широкого кола організацій.

Детальний розгляд та порівняльний аналіз підходів до захищеного віддаленого доступу — традиційного VPN, ZTNA та Software-Defined Perimeter — виконано в підрозділі 1.2.

1.2 Підходи до організації захищеного віддаленого доступу

Аналіз сучасного ландшафту загроз, проведений у підрозділі 1.1, демонструє, що традиційні механізми захисту периметру більше не відповідають вимогам корпоративної безпеки в умовах гібридної роботи та розподілених інфраструктур. Для обґрунтування архітектурних рішень даної роботи необхідно провести порівняльний аналіз трьох ключових підходів до організації захищеного віддаленого доступу: класичного VPN, Zero Trust Network Access (ZTNA) та Software-Defined Perimeter (SDP), — а також визначити роль IAM у контексті кожного з них.

Класичні VPN-рішення побудовані на основі моделі «замок із ровом» (castle-and-moat), де мережний периметр організації розглядається як захисна фортеця: усе, що перебуває всередині мережі, вважається надійним за замовчуванням, а зовнішній трафік блокується. Після успішної автентифікації VPN-клієнта користувач отримує широкий мережний доступ до корпоративних ресурсів — без

подальшого розмежування за принципом найменших привілеїв. Така архітектура передбачала, що загрози виключно зовнішні та не можуть виникати всередині периметру, що принципово суперечить реальному стану речей [16].

Зазначена модель породжує кілька критичних структурних уразливостей. По-перше, вона є вразливою до атак на основі скомпрометованих облікових даних: VPN-шлюз не здатний відрізнити легітимного користувача від атакуючого, що заволодів його обліковим записом, оскільки перевірка ідентичності відбувається одноразово — лише в момент встановлення з'єднання. Після проходження автентифікації сесія вважається довіреною до її завершення незалежно від подальших дій користувача та зміни контексту. Дослідження Fortinet показують, що у першій половині 2025 року зловмисники систематично використовували автентифіковані VPN-з'єднання для ручного горизонтального переміщення між вузлами мережі через вбудовані інструменти — RDP, SMB, WinRM — що суттєво ускладнює форензичний аналіз інцидентів [17].

По-друге, модель «замок із ровом» є неефективною у сучасних умовах через три структурні зміни корпоративного середовища:

- Масштабна віддалена робота — співробітники підключаються поза межами фізичного периметру, що розмиває концепцію «надійного внутрішнього простору»
- Міграція до хмари — застосунки та дані розподілені між хмарними платформами, а не сконцентровані в єдиному корпоративному дата-центрі
- Просунуті стійкі загрози (APT) — після потрапляння всередину мережі зловмисники безперешкодно здійснюють горизонтальне переміщення, оскільки традиційний VPN не забезпечує сегментацію на рівні застосунків

По-третє, ключовим структурним недоліком є статична модель довіри: рівень доступу визначається виключно на момент підключення і не переглядається динамічно залежно від зміни поведінки користувача, стану пристрою чи контексту сесії. Саме ця властивість робить VPN-рішення принципово несумісними з принципами сучасної корпоративної безпеки.

Наведені обмеження зумовили появу та стрімке поширення двох альтернативних концептуальних підходів — ZTNA та SDP, що розглядаються далі.

Zero Trust Network Access (ZTNA)

Zero Trust Network Access (ZTNA) — це концепція безпечного доступу, що базується на принципі «ніколи не довіряй, завжди перевіряй» (never trust, always verify), сформульованому аналітиком Forrester Research Джоном Кіндервагом у 2010 році. На відміну від класичної VPN-моделі, ZTNA надає доступ не до всієї корпоративної мережі, а виключно до конкретних авторизованих застосунків або ресурсів — причому лише за умови успішної перевірки ідентичності, стану пристрою та відповідності контекстуальним політикам безпеки [18].

Архітектурно ZTNA спирається на чотири ключові механізми верифікації, що застосовуються безперервно протягом усієї сесії:

- Верифікація ідентичності — підтвердження особи користувача через IAM-систему з підтримкою MFA
- Перевірка стану пристрою — валідація відповідності девайсу корпоративним політикам безпеки (наявність актуальних патчів, активний EDR-агент)
- Контекстуальна оцінка — аналіз географічного розташування, часу підключення та атипової поведінки
- Адаптивне управління доступом — динамічне обмеження або відкликання доступу в реальному часі при виявленні аномалій

Принципово важливо, що ZTNA є архітектурно залежним від IAM-системи: остання забезпечує централізовану ідентифікацію, видачу токенів доступу, управління ролями (RBAC) та реєстрацію всіх подій автентифікації, тоді як ZTNA виконує безперервну верифікацію на основі цих даних. Це підтверджується і в документах NIST: управління ідентифікацією є ключовим компонентом архітектури Zero Trust, оскільки реалізує принцип явної верифікації кожного запиту на доступ. Без централізованої IAM-платформи побудова справжнього ZTNA є неможливою, оскільки відсутній єдиний авторитетний джерело ідентичностей та політик доступу [19].

Software-Defined Perimeter (SDP)

Software-Defined Perimeter (SDP) — це архітектурний фреймворк інформаційної безпеки, розроблений Cloud Security Alliance (CSA) у 2013 році, що реалізує концепцію Zero Trust на мережевому рівні через динамічне формування ізольованих, персоналізованих мережевих сегментів. Ключовою відмінністю SDP від традиційного VPN є принцип «темної мережі» (dark network / black cloud): усі захищені ресурси — сервери, застосунки, сервіси — є повністю невидимими та недоступними для неавторизованих суб'єктів ще до проходження автентифікації. Це принципово відрізняється від класичного VPN, де сам факт існування та IP-адреса VPN-шлюзу є публічно відомими і доступними для зондування [20].

Технічною основою SDP є механізм Single Packet Authorization (SPA) — вдосконалена форма «port knocking», що реалізує принцип «автентифікація перш ніж підключення» (authenticate-first, connect-second). SPA-пакет є зашифрованим та підписаним однократним токеном, що містить ідентифікатор користувача, мітку часу та перелік запитуваних ресурсів. SDP-шлюз застосовує стратегію «deny-all» за замовчуванням: будь-який пакет, що не відповідає дійсному SPA-запиту, мовчки відкидається без відправки будь-якої відповіді. Таким чином, SDP-інфраструктура є стійкою до DDoS-атак, сканування портів та атак типу fingerprinting [21].

Архітектура SDP включає три ключових компоненти:

- SDP Controller — централізований вузол управління доступом, що взаємодіє з Identity Provider (IdP) для перевірки ідентичності та видачі динамічних політик доступу. У корпоративному контексті роль IdP виконує IAM-система, наприклад Keycloak
- SDP Gateway — точка примусового виконання політик (Policy Enforcement Point), що стоїть перед захищеними ресурсами та пропускає виключно авторизований трафік від автентифікованих клієнтів
- SDP Client — агент на пристрої користувача, що ініціює SPA-запит, отримує токен авторизації від Controller та встановлює шифрований канал зв'язку з Gateway

Зв'язок між SDP та IAM є структурним: SDP Controller делегує прийняття рішень про доступ Identity Provider, що надає підписані токени з переліком дозволених ресурсів і ролей. Таким чином, IAM є обов'язковим компонентом повноцінної SDP-реалізації, забезпечуючи ідентифікаційний базис для динамічного формування мережевих сегментів [22].

На основі проведеного аналізу у таблиці 1.1 узагальнено порівняльні характеристики трьох розглянутих підходів.

Таблиця 1.1

Порівняльний аналіз підходів до організації захищеного віддаленого доступу

Критерій	Традиційний VPN	ZTNA	SDP
Модель доступу	До всієї мережі після автентифікації	До конкретних застосунків	До конкретних ресурсів (динамічно)
Видимість ресурсів	Ресурси видимі після підключення	Прихований доступ до застосунків	Мережа повністю прихована до автентифікації
Модель довіри	Статична (одноразова перевірка)	Динамічна (безперервна верифікація)	Динамічна (автентифікація перш ніж з'єднання)
Верифікація сесії	Лише при вході	Безперервна (ідентичність + пристрій + поведінка)	Безперервна (SPA + токен-авторизація)
Залежність від IAM	Опціональна	Обов'язкова	Обов'язкова
Захист від lateral movement	Слабкий	Сильний (мікросегментація на рівні застосунку)	Сильний (100% невидимість неавторизованих ресурсів)
Захист від DDoS та сканування	Відсутній	Частковий	Сильний (deny-all за замовчуванням, SPA)
Гранулярність доступу	Мережевий рівень (груба)	Рівень застосунку	Рівень ресурсу/порту
Складність впровадження	Низька–середня	Середня	Висока
Придатність для відкритого ПЗ	Висока	Середня	Середня

Проведений аналіз демонструє, що як ZTNA, так і SDP усувають фундаментальний структурний недолік класичного VPN — статичну модель довіри з необмеженим мережевим доступом — через різні архітектурні механізми: ZTNA через безперервну верифікацію на рівні застосунку, SDP через «темну мережу» та

принцип «authenticate-first, connect-second». В обох підходах IAM виступає не допоміжним компонентом, а архітектурно необхідним базисом, без якого реалізація концепції Zero Trust є неможливою [23].

Таким чином, інтеграція WireGuard із централізованою IAM-платформою Keycloak, що реалізується в межах даної роботи, дозволяє відтворити ключові властивості ZTNA та SDP на базі відкритого програмного забезпечення: гранулярний доступ до ресурсів на основі RBAC-ролей, безперервну верифікацію ідентичності через токени OIDC, а також приховування серверних ресурсів завдяки «мовчазному» режиму WireGuard — де сервер не відповідає на непідтверджені пакети, що структурно відповідає принципу SPA в архітектурі SDP. Практична реалізація такого підходу на базі відкритого ПЗ розглядається в наступному підрозділі.

1.3 Порівняльний аналіз сучасних VPN-протоколів: WireGuard, OpenVPN та IPsec

З урахуванням виявлених обмежень класичного VPN та перспективності ZTNA-підходу, критично важливим є вибір конкретного транспортного протоколу для побудови захищеного тунелю.

Вибір протоколу VPN є одним із ключових архітектурних рішень при проєктуванні захищеної корпоративної мережі, оскільки він безпосередньо визначає рівень безпеки, продуктивність системи та складність її адміністрування. У сучасній практиці найбільш поширеними є три протоколи: IPsec, OpenVPN та WireGuard, кожен з яких має власну архітектурну концепцію, криптографічну базу та сферу оптимального застосування. У цьому підрозділі виконано їх детальний порівняльний аналіз з метою обґрунтування вибору для практичної реалізації в межах даної роботи.

IPsec

IPsec (Internet Protocol Security) — це набір протоколів і стандартів, розроблених IETF, що забезпечує захист IP-трафіку на мережевому рівні (L3 моделі OSI). Стандартизований у RFC 4301–4309, IPsec включає два основних режими

роботи: транспортний (захищається лише корисне навантаження пакета) та тунельний (захищається весь вихідний пакет, включаючи заголовки), що дозволяє його застосовувати як для point-to-point з'єднань, так і для site-to-site VPN. Ключовими компонентами IPsec є протоколи AH (Authentication Header) та ESP (Encapsulating Security Payload), а обмін ключами реалізується через протокол IKEv2 (RFC 7296).

З точки зору безпеки, IPsec підтримує гнучку криптографічну агільність: адміністратор може самостійно обирати алгоритми шифрування (AES-CBC, AES-GCM), хешування (SHA-256, SHA-384) та ключового обміну (DH group 14–21). Ця гнучкість є водночас і перевагою, і слабким місцем: неправильний вибір параметрів або застарілі конфігурації (наприклад, DES, MD5, group 1/2) суттєво знижують рівень захисту. Реалізація IPsec вимагає значного обсягу коду — сотні тисяч рядків у ядрі ОС — що збільшує поверхню атаки. Саме тому IPsec традиційно орієнтований на site-to-site сценарії в керованих корпоративних середовищах, де необхідна апаратна прискорення шифрування, а складність конфігурації виправдана масштабом [7].

OpenVPN

OpenVPN — це VPN-рішення з відкритим вихідним кодом, яке функціонує на прикладному рівні (L7) моделі OSI та реалізує захищене тунелювання через SSL/TLS. Розроблений у 2001 році Джеймсом Йонаном, протокол використовує бібліотеку OpenSSL як криптографічний бекенд, що забезпечує підтримку широкого спектру алгоритмів: шифрування (AES-256-GCM, ChaCha20), автентифікації (HMAC-SHA256) та обміну ключами (RSA, ECDH). OpenVPN може працювати як через UDP (рекомендовано для продуктивності), так і через TCP на будь-якому порту, включаючи порт 443, що дозволяє маскувати VPN-трафік під HTTPS та обходити більшість мережевих фільтрів [8].

Ключовою архітектурною особливістю OpenVPN є його робота у просторі користувача (userspace), а не в ядрі ОС. Це підвищує переносимість та гнучкість, але водночас спричиняє значні накладні витрати через постійне перемикання контексту між простором ядра і простором користувача при обробці кожного

пакета. Як наслідок, пропускна здатність OpenVPN суттєво нижча за WireGuard та апаратно-прискорений IPsec. Розмір кодової бази OpenVPN перевищує 70 000 рядків коду, що ускладнює аудит безпеки. Проте OpenVPN є зрілим, добре задокументованим рішенням з розвиненою екосистемою та широкою підтримкою клієнтів на всіх платформах [7].

WireGuard

WireGuard — це сучасний VPN-протокол, розроблений Джейсоном Донфельдом і представлений у 2016 році, інтегрований до ядра Linux починаючи з версії 5.6 (2020 рік). На відміну від попередників, WireGuard реалізований безпосередньо в ядрі ОС (~4 000 рядків коду) і базується на принципі «фіксованої криптографії» (cryptographic opinionatedness) [7]: протокол не надає адміністратору можливості обирати алгоритми, натомість жорстко визначає єдиний сучасний криптографічний стек:

- ChaCha20 — для симетричного шифрування даних (RFC 8439) ;
- Poly1305 — для автентифікації повідомлень (MAC) ;
- Curve25519 — для обміну ключами (ECDH) ;
- BLAKE2s — для хешування;
- HKDF — для виведення ключів.

Обмін ключами у WireGuard базується на фреймворку Noise Protocol, що реалізує 1-RTT handshake з властивостями forward secrecy та захисту від атак відтворення (replay protection). Принципово важливою особливістю є концепція «мовчазного» (silent) однорангового вузла: WireGuard не відповідає на непідтверджені пакети, що унеможливлює сканування порту сервера та суттєво ускладнює атаки типу DDoS і fingerprinting [9].

Завдяки реалізації в ядрі та оптимізованим криптографічним примітивам, WireGuard демонструє пропускну здатність до 9,5 Гбіт/с на сучасному обладнанні — значно вище за OpenVPN та IPsec у порівнюваних умовах. Крім того, встановлення з'єднання WireGuard відбувається майже миттєво (<100 мс), тоді як IPsec IKEv2 потребує кількох секунд на повторне встановлення тунелю [10].

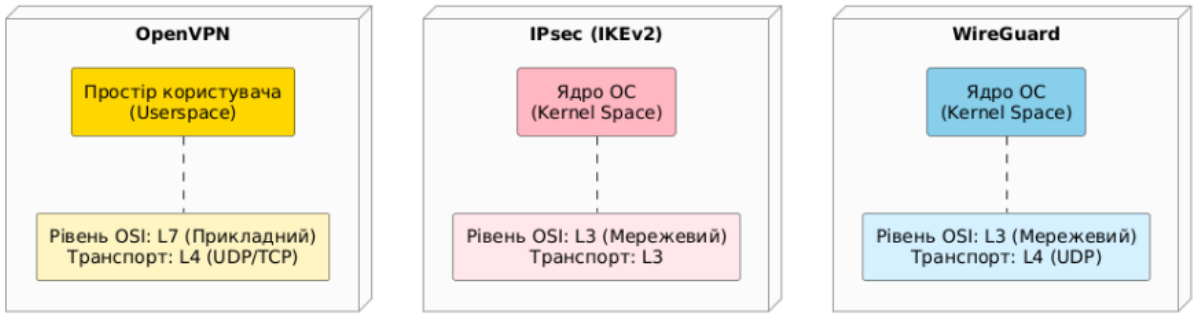


Рисунок 1.1 — Порівняльна діаграма архітектур протоколів

Таблиця 1.2

Порівняльний аналіз VPN-протоколів

Критерій	IPsec (IKEv2)	OpenVPN	WireGuard
Рівень OSI	L3 (мережевий)	L4–L7 (транспортний / прикладний)	L3 (мережевий)
Реалізація	Ядро ОС	Простір користувача	Ядро ОС
Розмір кодової бази	~100 000+ рядків	~70 000+ рядків	~4 000 рядків
Шифрування	AES-GCM, AES-CBC (конфігурується)	AES-256-GCM, ChaCha20 (конфігурується)	ChaCha20-Poly1305 (фіксовано)
Обмін ключами	IKEv1/IKEv2 (Diffie-Hellman)	TLS (RSA/ECDH)	Noise Protocol (Curve25519)
Пропускна здатність	Висока (з апаратним прискоренням)	Середня	Дуже висока
Затримка handshake	1–3 с (IKEv2)	3–10 с (TLS)	< 100 мс
Проходження NAT	Потребує NAT-T (UDP 4500)	Відмінне (порт 443 TCP/UDP)	Відмінне (UDP, будь-який порт)
Складність конфігурації	Висока	Середня	Низька
Підтримка клієнтів	Вбудована у Windows/iOS/Android	Широка (всі платформи)	Широка (всі платформи)
Аудитованість	Складна	Помірна	Проста (мала кодова база)
Вразливість до помилок конфігурації	Висока (гнучкість = ризик)	Середня	Низька (фіксована криптографія)

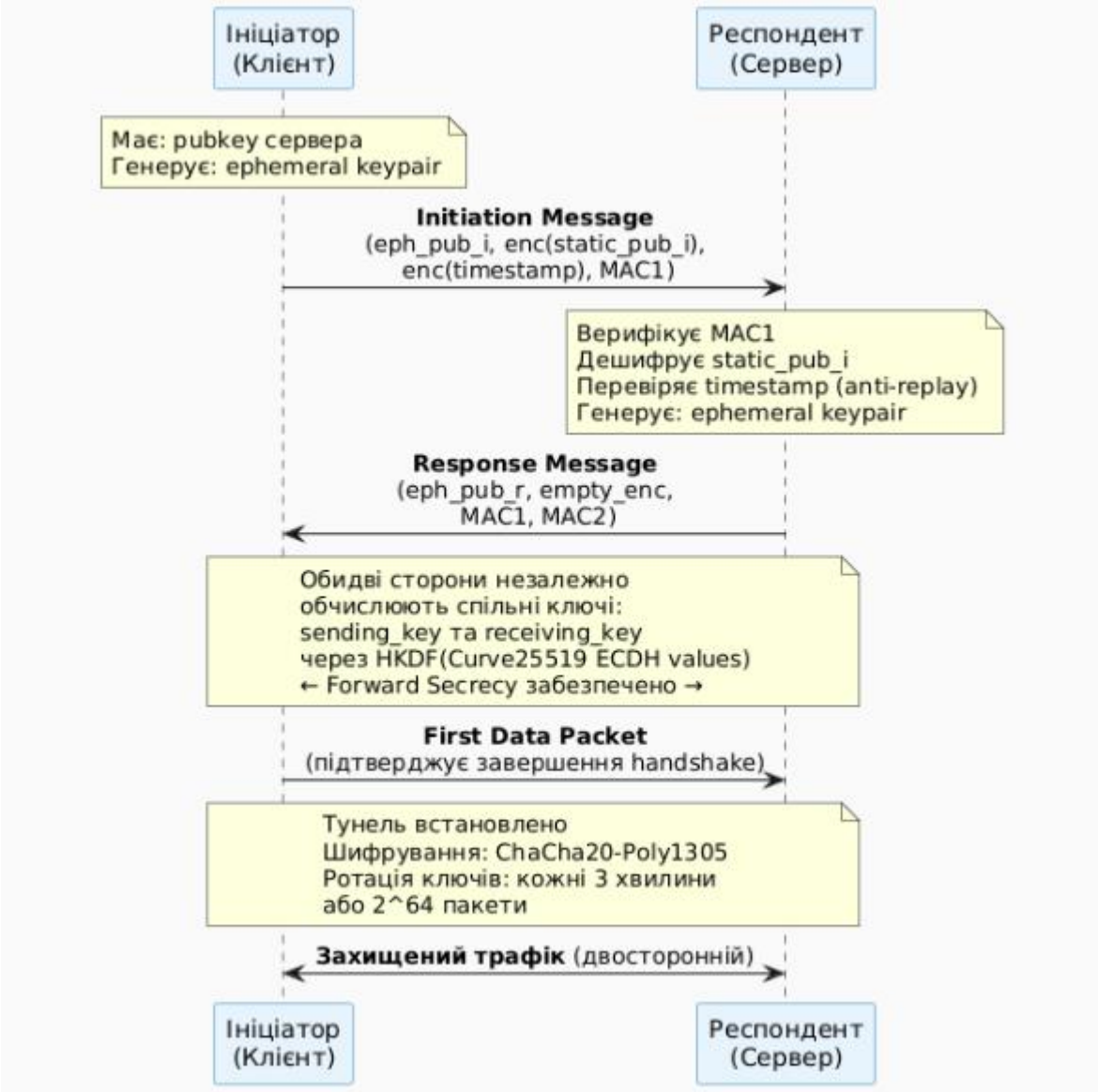


Рисунок 1.2 — Діаграма процесу WireGuard Handshake

За результатами проведеного аналізу, для реалізації VPN-інфраструктури в межах даної роботи обрано протокол WireGuard з таких підстав:

1. Мінімальна поверхня атаки — ~4 000 рядків коду проти 70 000+ у OpenVPN та 100 000+ у IPsec робить аудит безпеки практично здійсненним

2. Фіксована сучасна криптографія — унеможливлює вибір слабких алгоритмів внаслідок помилки конфігурації, що критично важливо для корпоративного середовища

3. Найвища продуктивність — реалізація в ядрі ОС забезпечує пропускну здатність до 9,5 Гбіт/с та затримку handshake менше 100 мс

4. Простота конфігурації та адміністрування — зменшує операційні витрати та ризик помилок налаштування

5. Вбудована підтримка у Linux kernel 5.6+ — не потребує сторонніх модулів, що підвищує стабільність і підтримку в довгостроковій перспективі

IPsec виключено через надмірну складність конфігурації та підвищений ризик помилок при ручному налаштуванні криптографічних параметрів. OpenVPN виключено через значно нижчу продуктивність через userspace-архітектуру та більший розмір кодової бази. Водночас, обидва протоколи враховані в порівняльному аналізі як релевантні альтернативи, що широко застосовуються у галузі.

1.4 Дослідження архітектури та можливостей сучасних IAM-систем на прикладі Keycloak

1.4.1 Концепція IAM та її роль у корпоративній безпеці

Identity and Access Management (IAM) — це комплексна дисципліна та клас програмних рішень, що забезпечує централізоване управління цифровими ідентичностями користувачів і визначає умови, за яких ці ідентичності отримують доступ до корпоративних ресурсів. Відповідно до моделі NIST SP 800-63, система IAM охоплює чотири фундаментальні функції: ідентифікацію (встановлення того, хто є суб'єкт), автентифікацію (підтвердження заявленої ідентичності), авторизацію (визначення дозволених дій) та аудит (реєстрацію всіх подій доступу). У межах дослідження концепції Zero Trust, IAM є ключовим компонентом, оскільки реалізує принцип «ніколи не довіряй, завжди перевіряй» — кожен запит на доступ до ресурсу має бути явно автентифікований та авторизований незалежно від розташування суб'єкта в мережі [11].

Без централізованої IAM-системи управління доступом у великих організаціях перетворюється на розрізнений набір локальних механізмів автентифікації у кожному застосунку. Такий підхід призводить до дублювання облікових записів, неузгодженості політик, складнощів з відкликанням доступу звільнених співробітників та відсутності єдиного аудитного сліду. Централізована IAM вирішує ці проблеми, виступаючи єдиною точкою правди (Single Source of Truth) щодо ідентичностей та прав доступу [12].

1.4.2 Ключові протоколи та стандарти IAM

Сучасні IAM-системи базуються на трьох основних стандартах, що визначають механізми автентифікації та авторизації у розподілених середовищах.

OAuth 2.0 (RFC 6749) — це фреймворк авторизації, що дозволяє застосунку отримати обмежений делегований доступ до ресурсів від імені користувача без передачі облікових даних. OAuth 2.0 оперує поняттями Authorization Server, Resource Server та Client, і визначає кілька сценаріїв (flows): Authorization Code, Client Credentials, Implicit (застарілий) та Device Authorization. Важливо розуміти, що OAuth 2.0 є протоколом авторизації, але не автентифікації — він підтверджує права, але не ідентичність [13].

OpenID Connect (OIDC) — це рівень автентифікації, побудований поверх OAuth 2.0 (специфікація OpenID Foundation). OIDC додає поняття ID Token — підписаного JWT-токена, що містить заявки (claims) про ідентичність користувача: sub (унікальний ідентифікатор), email, name, а також дані про час автентифікації та використаний метод [15]. Таким чином, OIDC відповідає на питання «хто є цей користувач?», тоді як OAuth 2.0 — «що цьому користувачу дозволено?».

SAML 2.0 (Security Assertion Markup Language) — це XML-базований стандарт для обміну даними автентифікації та авторизації між Identity Provider (IdP) та Service Provider (SP). SAML широко застосовується в корпоративному середовищі, особливо для інтеграції з legacy-системами та застосунками Microsoft. На відміну від OIDC, SAML є більш громіздким у реалізації, але забезпечує зворотню сумісність з великою кількістю enterprise-застосунків [13].

1.4.3 Огляд IAM-рішень та обґрунтування вибору Keycloak

На ринку представлено широкий спектр IAM-платформ — від хмарних SaaS-рішень до on-premise систем з відкритим вихідним кодом.

Таблиця 1.3

Порівняльний аналіз IAM-платформ

Критерій	Keycloak	Okta	Azure Active Directory	Authentik
Модель розповсюдження	Open-source (Apache 2.0)	SaaS (комерційний)	SaaS / Hybrid (комерційний)	Open-source
Розгортання	Self-hosted / Cloud	Cloud-only	Cloud / Hybrid	Self-hosted
Вартість	Безкоштовно	$2–$15/міс на користувача	Від $6/міс на користувача	Безкоштовно
OAuth 2.0 / OIDC	+	+	+	+
SAML 2.0	+	+	+	+
RBAC / ABAC	+	+	+ (обмежено)	+
LDAP / AD інтеграція	+	+	Вбудовано	+
MFA	+	+	+	+
Контроль над даними	Повний	Обмежений	Обмежений	Повний

Для реалізації в межах даної роботи обрано Keycloak — open-source IAM-платформу, що розробляється компанією Red Hat і є частиною екосистеми CNCF (Cloud Native Computing Foundation). Ключові підстави для вибору: повний контроль над інфраструктурою та даними, відсутність вартості ліцензування, підтримка всіх необхідних стандартів, а також активна спільнота та розвинена документація.

1.4.4 Архітектура Keycloak

Keycloak побудований на основі фреймворку Quarkus і являє собою повноцінний Identity Provider (IdP), що реалізує всі функції IAM-системи в єдиному сервісі. Загальну архітектуру Keycloak та його взаємодію з компонентами корпоративної інфраструктури наведено на рисунку 1.3. [14]:

Realm — це ізольований простір конфігурації, що містить усіх користувачів, ролі, клієнтів та налаштування безпеки. Один екземпляр Keycloak може містити кілька realm, що дозволяє розділяти різні організації або середовища (development, staging, production) на одному сервері.

Client — це застосунок або сервіс, зареєстрований у Keycloak, що може запитувати автентифікацію користувачів або отримувати токени доступу. Клієнти бувають confidential (з секретом, для backend-сервісів) та public (без секрету, для SPA та мобільних застосунків).

User Federation — механізм підключення зовнішніх сховищ ідентичностей. Keycloak підтримує синхронізацію з LDAP та Microsoft Active Directory, що забезпечує безшовну інтеграцію в наявну корпоративну інфраструктуру.

Identity Brokering — дозволяє Keycloak виступати посередником між зовнішніми IdP (Google, GitHub, Microsoft Entra ID) та застосунками, забезпечуючи федеративну автентифікацію без зміни застосунків.

Authentication Flows — гнучко налаштовувані сценарії автентифікації. Адміністратор може конфігурувати послідовність кроків: введення пароля, OTP, WebAuthn/Passkey, умовна MFA тощо.

Token Management — Keycloak видає три типи токенів: Access Token (JWT з правами доступу, короткоживучий — 5 хвилин за замовчуванням), Refresh Token (для отримання нового Access Token без повторної автентифікації) та ID Token (для підтвердження ідентичності в OIDC-потоці).

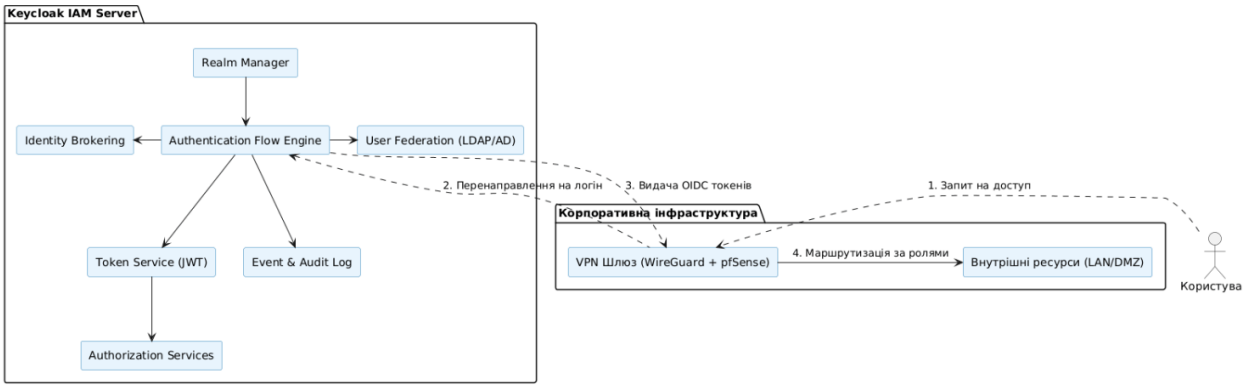


Рисунок 1.3 — Архітектура Keycloak та взаємодія з корпоративними компонентами

Процес автентифікації користувача при підключенні до VPN-системи реалізується через OIDC Authorization Code Flow. Послідовність взаємодії компонентів системи в процесі автентифікації наведено на рисунку 1.4.

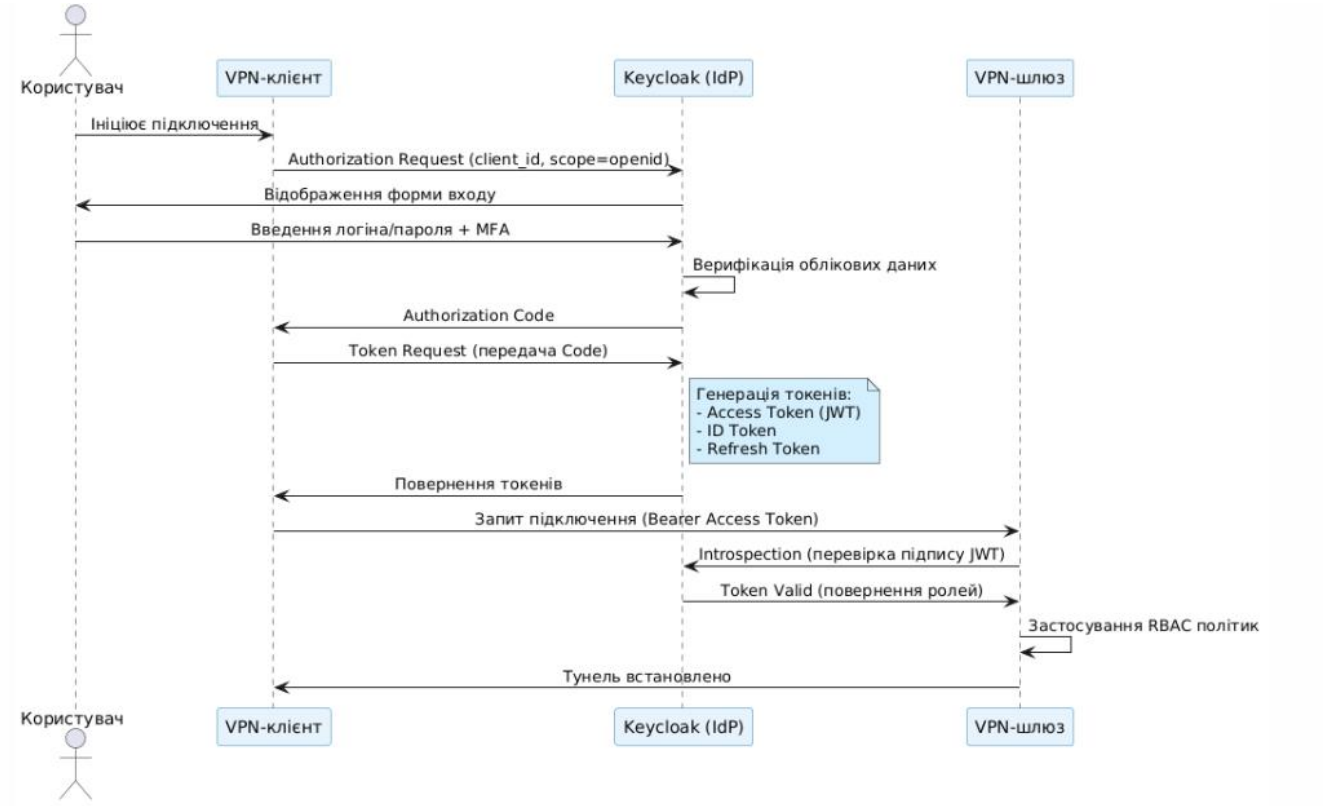


Рисунок 1.4 — OIDC Authorization Code Flow у корпоративній VPN-системі

Keycloak підтримує два рівні ролей: Realm Roles (глобальні для всього realm) та Client Roles (специфічні для конкретного клієнта/застосунку). Ролі можуть об'єднуватись у Composite Roles — ієрархічні групи, де одна роль успадковує права іншої. Ієрархію ролей для розроблюваної системи наведено на рисунку 1.5.



Рисунок 1.5 — Ієрархія ролей у Keycloak для корпоративної VPN-системи

Серед ключових можливостей Keycloak, що мають безпосереднє значення для реалізації даної роботи, виділяють:

- Single Sign-On (SSO) — одноразова автентифікація надає доступ до всіх зареєстрованих клієнтів у межах realm без повторного введення облікових даних
- Multi-Factor Authentication (MFA) — підтримка TOTP (Google Authenticator), WebAuthn/FIDO2 та SMS-кодів на рівні realm або для конкретних ролей
- Brute Force Protection — автоматичне блокування облікового запису після визначеної кількості невдалих спроб входу
- Event Logging — вбудоване журналювання подій автентифікації (успішний вхід, невдала спроба, logout, видача токена) з можливістю передачі до зовнішніх SIEM-систем через Event Listener SPI
- Fine-Grained Authorization (UMA 2.0) — розширена авторизація на рівні ресурсів, що дозволяє задавати умови доступу на основі атрибутів контексту (час, IP-адреса, пристрій)
- High Availability — кластеризація через Infinispan та підтримка multi-site deployments для забезпечення відмовостійкості

Таким чином, Keycloak є комплексним рішенням, що закриває всі вимоги до централізованої IAM-підсистеми в рамках корпоративної VPN-інфраструктури: забезпечує стандартизовану автентифікацію через OIDC, гнучке рольове розмежування доступу, повноцінний аудит подій та готову інтеграцію з мережевим обладнанням і сервісами через стандартні протоколи.

1.5 Вибір інструментальних засобів та формування вимог до апаратно-програмного комплексу

Для практичної реалізації та дослідження корпоративної VPN-інфраструктури необхідне середовище, яке дозволяє відтворити реальну мережеву взаємодію між компонентами системи без використання фізичного мережевого обладнання. З-поміж наявних інструментів мережевої симуляції (Cisco Packet Tracer, EVE-NG, GNS3) для виконання практичної частини роботи обрано графічний симулятор мережі GNS3 (Graphical Network Simulator-3).

На відміну від Cisco Packet Tracer, який є симулятором з обмеженим набором команд, GNS3 є емулятором, що дозволяє запускати повноцінні образи операційних систем через гіпервізори QEMU/KVM та інтегрувати Docker-контейнери [15]. Це є принциповою вимогою для даної роботи, оскільки всі ключові вузли розроблюваної системи (pfSense, WireGuard, Keycloak) потребують розгортання в середовищі реальних ОС. На відміну від EVE-NG, GNS3 розповсюджується за відкритою ліцензією та має широку базу готових шаблонів (appliances) для швидкого розгортання мережевих вузлів.

Вимоги до апаратного забезпечення Враховуючи архітектурну складність проєктованої системи та використання технології контейнеризації для IAM-платформи, до апаратного забезпечення хост-машини (фізичного ПК, на якому виконується моделювання) висуваються підвищені вимоги:

- Процесор (CPU): Підтримка апаратної віртуалізації (Intel VT-x або AMD-V), мінімум 4 фізичні ядра (рекомендовано 8 потоків) для паралельної обробки трафіку віртуальними машинами.
- Оперативна пам'ять (RAM): Мінімум 16 ГБ загальної оперативної пам'яті. Розподіл ресурсів для GNS3 VM: не менше 8 ГБ RAM (з яких 2 ГБ виділяється для контейнера Keycloak, 1 ГБ для шлюзу pfSense, та по 1-2 ГБ для клієнтських систем на базі Linux/Windows).
- Дискова підсистема: SSD-накопичувач з мінімум 40 ГБ вільного простору для зберігання образів віртуальних машин та забезпечення високої швидкості операцій вводу/виводу (IOPS) при роботі з базами даних (PostgreSQL).

На основі проведеного аналізу для реалізації системи обрано наступний стек програмного забезпечення, зведений у таблицю 1.4.

Таблиця 1.4

Програмні компоненти апаратно-програмного комплексу

Компонент	Програмне забезпечення	Призначення
Середовище моделювання	GNS3 (v. 2.2.x)	Візуальне проєктування та емуляція топології мережі
Гіпервізор хосту	Oracle VirtualBox 7.0+	Виконавче середовище для GNS3 VM та клієнтських ОС
Міжмережевий екран	pfSense CE 2.7+	Маршрутизація, фільтрація трафіку, сегментація мережі
VPN-протокол	WireGuard (вбудований у ядро Linux)	Забезпечення захищеного тунелювання (L3)
VPN-портал	Firezone 0.7.x	Веб-платформа управління WireGuard-з'єднаннями з OIDC-інтеграцією до Keycloak
IAM-система	Keycloak 24+	Централізована автентифікація (OIDC) та авторизація
СУБД для IAM	PostgreSQL 16	Зберігання конфігурацій, користувачів та політик Keycloak
Контейнеризація	Docker + Docker Compose	Ізольоване розгортання Keycloak та СУБД
Базова ОС вузлів	Ubuntu Server 22.04 LTS	Операційна система серверних та клієнтських вузлів

На основі аналізу сучасних кіберзагроз та дослідження технологічних рішень сформовано базові вимоги до проєктованого апаратно-програмного комплексу. Головним завданням системи є забезпечення захищеного віддаленого доступу до корпоративних ресурсів на базі протоколу WireGuard із централізованою автентифікацією через платформу Keycloak за стандартом OpenID Connect. Мережева інфраструктура повинна підтримувати рольову модель контролю доступу (RBAC), де маршрутизація та блокування несанкціонованого трафіку між сегментами суворо контролюються міжмережевим екраном pfSense. Крім того, система має вести детальний аудит подій доступу, бути відтворюваною в межах віртуалізованих середовищ та будуватися виключно на актуальних версіях відкритого програмного забезпечення без критичних вразливостей.

Сформовані вимоги є базисом для проєктування архітектури системи (Розділ 2) та критеріями для оцінки її ефективності під час тестування (Розділ 3).

Висновки до першого розділу

Аналіз сучасного стану загроз підтверджує, що традиційні VPN-рішення зі статичною моделлю довіри є принципово недостатніми для захисту розподілених корпоративних інфраструктур в умовах зростаючих кіберзагроз. Порівняльний аналіз ZTNA та SDP показує, що обидва підходи усувають ключовий недолік класичного VPN через динамічну верифікацію ідентичності та мікросегментацію, а централізована IAM-система є архітектурно необхідним компонентом реалізації Zero Trust. Серед розглянутих протоколів WireGuard є оптимальним вибором завдяки мінімальній кодовій базі (~4 000 рядків), фіксованому криптографічному стеку на основі Noise Protocol та найвищій продуктивності. Keycloak як open-source IAM-платформа з підтримкою OIDC, OAuth 2.0 та RBAC забезпечує всі необхідні функції централізованої автентифікації та аудиту.

На підставі проведеного аналізу обрано стек для практичної реалізації: WireGuard/Firezone як VPN-шлюз, Keycloak 24 як IAM-платформа та pfSense CE 2.7 як засіб мережевої сегментації, розгорнуті у середовищі GNS3. Ця комбінація відкритого програмного забезпечення відтворює ключові властивості ZTNA-архітектури без комерційних ліцензій та є основою для проєктування системи в Розділі 2.

РОЗДІЛ 2. ПРОЄКТУВАННЯ КОРПОРАТИВНОЇ VPN-ІНФРАСТРУКТУРИ З ІНТЕГРАЦІЄЮ KEYCLOAK

2.1 Розробка загальної архітектури та топології захищеної мережі

Проєктована система захищеного доступу базується на трирівневій архітектурній моделі, в якій кожен рівень виконує чітко визначену функцію безпеки. Перший рівень — периметровий захист — реалізується міжмережевим екраном pfSense, який контролює весь вхідний та вихідний трафік і забезпечує сегментацію мережі. Другий рівень — захищений транспорт — забезпечується VPN-шлюзом Firezone на базі протоколу WireGuard, що встановлює зашифровані тунелі між віддаленими клієнтами та корпоративною мережею. Третій рівень — централізована ідентифікація та авторизація — реалізується платформою Keycloak, яка виступає єдиним джерелом правди щодо ідентичностей користувачів та їх прав доступу. Така багаторівнева модель відповідає принципу Defense in Depth — жоден окремий скомпрометований компонент не призводить до повного порушення безпеки системи.

В основу архітектури покладено концепцію Zero Trust: кожне підключення до корпоративних ресурсів вимагає явної криптографічної автентифікації через Firezone та ідентифікації через Keycloak незалежно від розташування клієнта, а обсяг наданого мережевого доступу суворо обмежується роллю користувача на рівні правил pfSense. Навіть у разі компрометації тунелю одного клієнта зловмисник отримає доступ лише до тих ресурсів, що явно дозволені відповідною роллю — горизонтальне переміщення між сегментами унеможливлюється.

Структура мережевих сегментів

Мережа розподілена на логічні сегменти, кожен з яких ізольований на рівні pfSense. Загальна топологія стенду включає дві маршрутизовані зони: корпоративну (LAN) та клієнтську (Remote LAN), з'єднані через спільний транзитний WAN-сегмент, що емулюється засобами GNS3 NAT Cloud та двома маршрутизаторами Cisco 7200.

Загальну топологію проєктованої системи наведено на рисунку 2.1.

Таблиця 2.1

Мережева адресація та призначення сегментів

Сегмент	Вузол / Інтерфейс	Підмережа	IP-адреса
WAN (GNS3 NAT)	GNS3 NAT Cloud (шлюз)	10.0.3.0/24	10.0.3.2
WAN R1	R1 Cisco 7200 fa0/0	10.0.3.0/24	10.0.3.16
Transit R1–pfSense	R1 fa1/0 ↔ pfSense em0	10.1.1.0/30	R1: 10.1.1.1; pfSense: 10.1.1.2
LAN (Corporate)	pfSense em1	10.0.0.0/24	10.0.0.1
VPN-підмережа	Firezone wg0	10.8.0.0/24	10.8.0.1
Remote LAN	R2 Cisco 7200 fa1/0	172.16.0.0/24	172.16.0.1

Сегмент LAN містить усі серверні вузли: Firezone WireGuard-сервер (10.0.0.2), Keycloak IAM-сервер (10.0.0.3) та внутрішній корпоративний сервер (10.0.0.10). Клієнти підключаються через WireGuard і отримують динамічно призначувані адреси у підмережі 10.8.0.0/24, після чого їх трафік маршрутизується через pfSense відповідно до призначеної ролі.

Таблиця 2.2

Адресація серверних вузлів корпоративного сегменту

Вузол	IP-адреса	Сервіс	Порт
pfSense (LAN-шлюз)	10.0.0.1	Firewall / NAT	—
Firezone	10.0.0.2	WireGuard VPN + веб-портал	UDP 51820, TCP 80
Keycloak	10.0.0.3	OIDC / SSO	TCP 8080
Internal Server	10.0.0.10	Корпоративний веб-ресурс	TCP 80

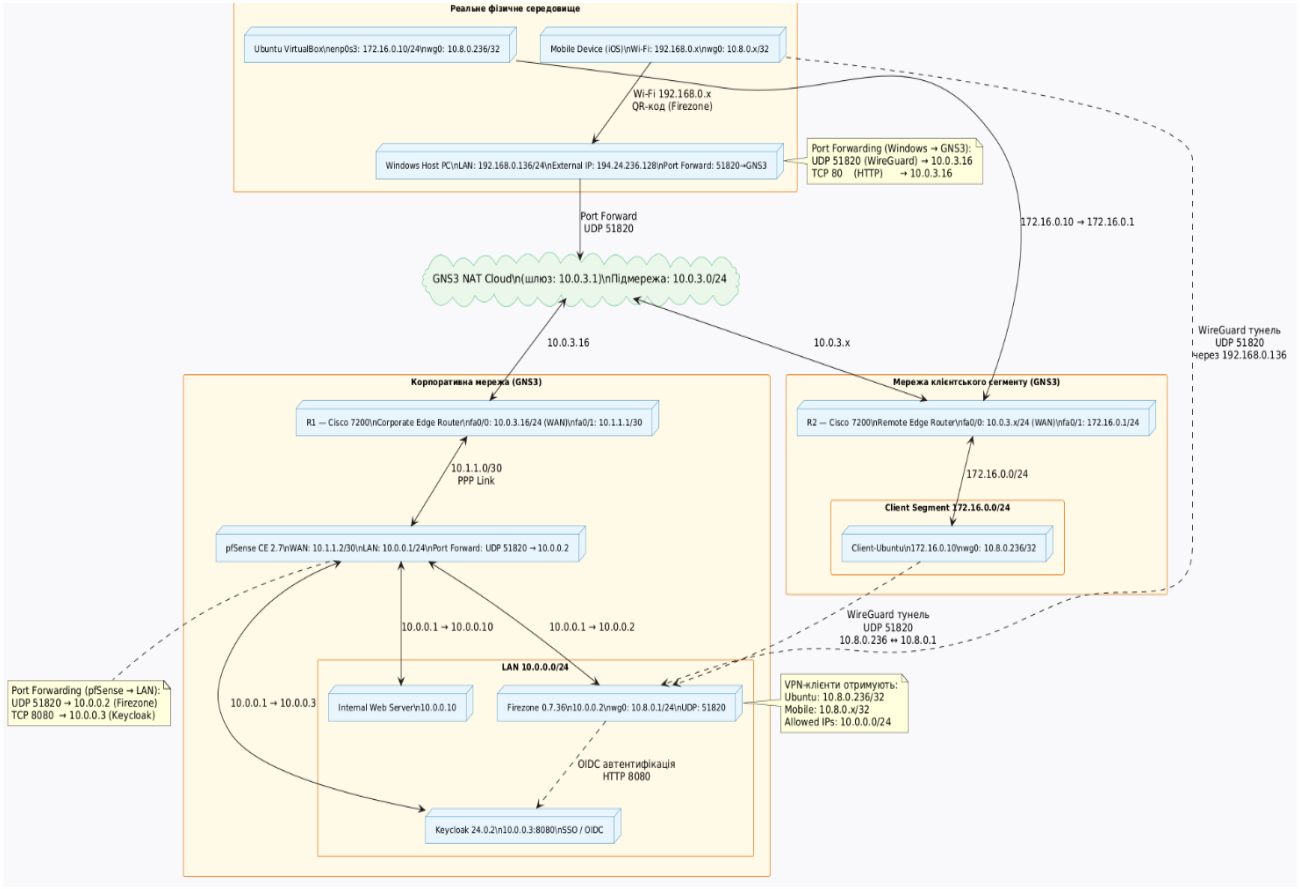


Рисунок 2.1 — Загальна топологія захищеної мережі

Система підтримує дві ролі користувачів, що визначають обсяг дозволеного мережевого доступу. Оскільки Firezone призначає кожному клієнту індивідуальну IP-адресу у підмережі 10.8.0.0/24, pfSense однозначно ідентифікує роль клієнта за адресою джерела трафіку та застосовує відповідну політику фільтрації.

Таблиця 2.3

Матриця доступу за ролями

Роль	Тунельна IP	Internal Server (10.0.0.10)	Keycloak (10.0.0.3)	pfSense панель (10.0.0.1)	Інші сегменти
vpn-user	10.8.0.x/32	TCP 80	TCP 8080 (лише вхід)	Заблоковано	-
admin	10.8.0.x/32	Повний (TCP 80, 443, 22)	Повний (TCP 8080)	TCP 443	-

Обидві ролі не мають доступу до WAN-сегменту (10.0.3.0/24) та мережі R2-Remote (172.16.0.0/24) — ці сегменти знаходяться поза корпоративною зоною і не є цільовими ресурсами для VPN-клієнтів. Відповідна заборона забезпечується правилом Default Deny №6 pfSense та ACL OUTSIDE-IN на R1.

Такий підхід реалізує принцип найменших привілеїв (Principle of Least Privilege): кожен користувач отримує рівно той обсяг мережевого доступу, що необхідний для виконання його функцій. Прив'язка ролей до IP-адрес у VPN-підмережі є ключовим архітектурним рішенням, що дозволяє застосовувати політики безпеки на рівні stateful firewall-правил pfSense без необхідності глибокої інспекції пакетів (DPI).

Оскільки ядро протоколу WireGuard не підтримує інтерактивну HTTP-автентифікацію на рівні самого протоколу, взаємодія компонентів побудована за принципом поділу фази ідентифікації та фази встановлення тунелю. Firezone виступає посередником між Keycloak та WireGuard: клієнт спочатку автентифікується через OIDC-портал Firezone (інтегрований з Keycloak), після успішної автентифікації Firezone реєструє публічний ключ клієнта у конфігурацію WireGuard та призначає тунельну IP-адресу — і лише тоді клієнт виконує криптографічний WireGuard Handshake та отримує доступ до мережі.

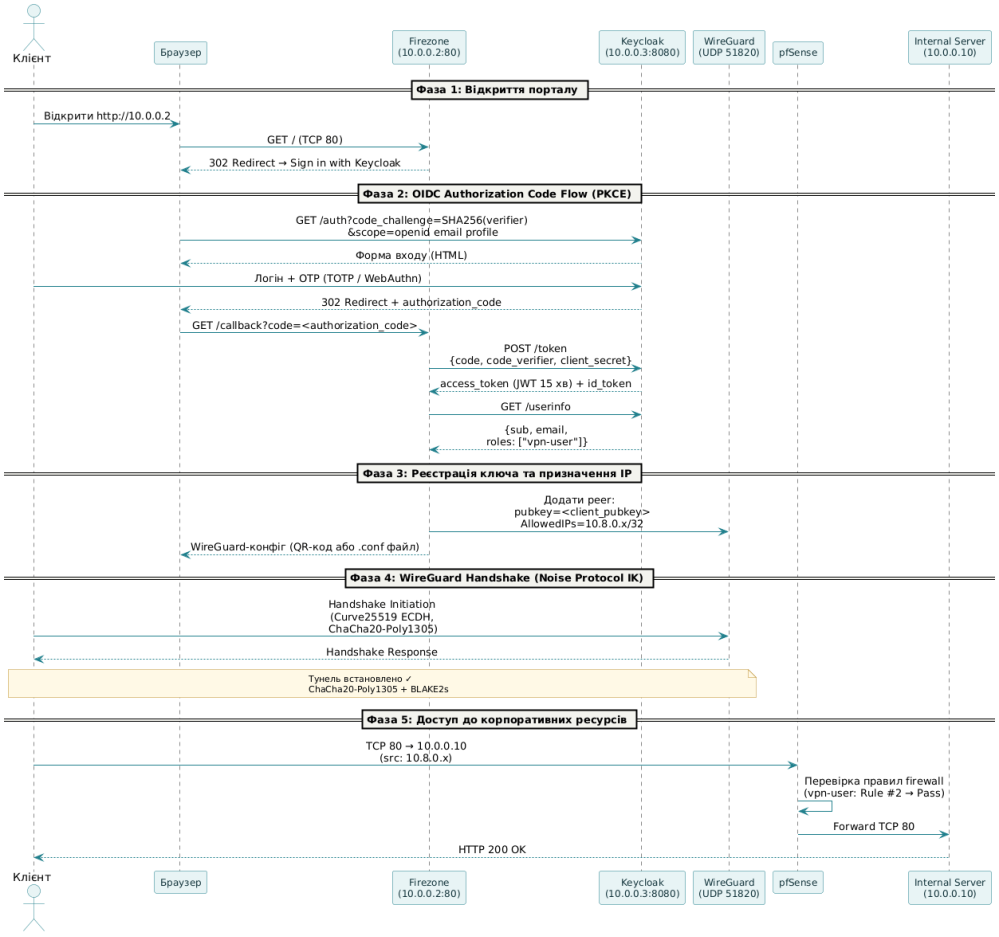


Рисунок 2.2 — Послідовність взаємодії компонентів при підключенні клієнта

Прийняті архітектурні рішення обґрунтовуються такими чинниками:

Розміщення Keycloak та Firezone у внутрішньому LAN забезпечує захист обох сервісів брандмауером pfSense. Для забезпечення первинного доступу зовнішніх клієнтів до порталу автентифікації pfSense публікує два сервіси через механізм Port Forward: веб-портал Firezone (TCP 80 → 10.0.0.2) для ініціювання OIDC-сесії та OIDC-ендпойнти Keycloak (TCP 8080 → 10.0.0.3) для обробки автентифікаційних перенаправлень браузера. Адміністративні панелі обох сервісів залишаються доступними виключно через WireGuard-тунель.

Виділення окремої VPN-підмережі (10.8.0.0/24) дозволяє застосовувати гранульовані правила фільтрації за IP-адресою джерела без перетину адресних просторів корпоративної та клієнтської мереж.

Один WireGuard-сервер як єдина точка входу для всіх клієнтів спрощує аудит та моніторинг з'єднань — весь зовнішній доступ до корпоративної мережі проходить через одну керовану точку, що є характерною ознакою ZTNA-архітектури.

Підтримка гетерогенних клієнтів: архітектура не прив'язана до конкретного пристрою чи ОС — WireGuard-клієнт наявний на Windows, Linux, macOS, Android та iOS, що забезпечує сумісність з BYOD-сценаріями.

Масштабованість: додавання нових клієнтів вимагає лише реєстрації нового пристрою у Firezone та призначення ролі у Keycloak — без зміни правил pfSense або мережевої адресації, оскільки всі ролі та відповідні IP-діапазони вже визначені на рівні архітектури.

2.2 Проєктування підсистеми віддаленого доступу на базі WireGuard та міжмережевого екрана pfSense

Підсистема віддаленого доступу є центральним елементом проєктованої інфраструктури, оскільки саме вона забезпечує захищений канал між зовнішніми клієнтами та корпоративними ресурсами. Підсистема складається з трьох нерозривно пов'язаних компонентів: платформи управління VPN Firezone, що відповідає за централізовану реєстрацію ключів та адміністрування тунелів;

протоколу WireGuard, що забезпечує шифрування транспортного рівня; та міжмережевого екрана pfSense, що реалізує маршрутизацію трафіку, міжсегментну фільтрацію та рольові політики доступу. Розподіл відповідальності між компонентами є принциповим архітектурним рішенням: Firezone вирішує задачу «хто і як отримує ключ для підключення», WireGuard — «як безпечно доставити трафік», pfSense — «куди цей трафік допустити».

Пряме використання WireGuard вимагає ручного управління публічними ключами клієнтів через конфігураційні файли, що є практично непридатним для корпоративного середовища з динамічним складом користувачів. Тому в проєктованій системі застосовується платформа Firezone 0.7.x — open-source рішення, що надає веб-інтерфейс для централізованого управління WireGuard-тунелями та нативну інтеграцію з OIDC-сумісними провайдерами ідентичностей.

Firezone виконує такі функції в системі:

- автоматично генерує пари ключів Curve25519 для кожного нового пристрою;
- динамічно призначає тунельні IP-адреси з підмережі 10.8.0.0/24 та реєструє їх у конфігурацію WireGuard як AllowedIPs = <клієнтська_IP>/32;
- генерує QR-коди та .conf-файли для мобільних і десктопних клієнтів;
- інтегрується з Keycloak через OIDC для автентифікації користувачів перед видачею VPN-конфігурацій.

WireGuard-сервер Firezone прослуховує порт UDP 51820 на адресі 10.0.0.2, а веб-портал управління доступний за адресою http://10.0.0.2 (TCP 80). Доступ до порталу захищений автентифікацією через Keycloak, що описано у підрозділі 2.3. Завдяки такій архітектурі WireGuard-ключі автоматично відкликаються при деактивації облікового запису в Keycloak — що усуває ключову проблему класичного VPN: необхідність ручного видалення peer-конфігурацій при звільненні або блокуванні співробітника.

WireGuard використовує модель ідентифікації вузлів на основі криптографічних публічних ключів Curve25519, а не IP-адрес. Кожен вузол має унікальну пару ключів: приватний ключ зберігається локально і ніколи не

передається мережею, публічний ключ реєструється сервером Firezone як ідентифікатор однорангового вузла (peer). Firezone призначає кожному пристрою фіксовану тунельну IP-адресу з підмережі 10.8.0.0/24 та записує її у конфігурацію WireGuard. Адресацію вузлів WireGuard-підмережі наведено у таблиці 2.4.

Таблиця 2.4

Адресація вузлів WireGuard-підмережі

Вузол	Роль	Тунельна IP (wg0)	AllowedIPs (сервер)	Endpoint клієнта
Firezone (сервер)	Сервер	10.8.0.1/24	0.0.0.0/0	—
Client-Ubuntu	vpn-user	динамічно /32	<client-ip>/32	10.0.3.16:51820
Client-Mobile (iOS)	vpn-user	динамічно /32	<client-ip>/32	10.0.3.16:51820
Client-Admin	admin	динамічно /32	<client-ip>/32	10.0.3.16:51820

Параметр AllowedIPs на стороні сервера виконує подвійну функцію: визначає, які IP-адреси дозволено використовувати кожному peer — це захищає від IP-спуфінгу — та задає маршрути для зворотного трафіку. На стороні клієнта AllowedIPs = 10.0.0.0/24 забезпечує split tunneling: через тунель маршрутизується лише трафік до корпоративної підмережі, решта трафіку клієнта іде безпосередньо в Інтернет. Це рішення обґрунтоване тим, що повне тунелювання є зайвим для корпоративного сценарію та збільшує навантаження на сервер.

Маршрутизація трафіку через R1 та pfSense

Зовнішній клієнт не має прямого доступу до WAN-інтерфейсу pfSense — між клієнтом і периметром корпоративної мережі розташовані два маршрутизатори Cisco 7200 (R1 та R2), що утворюють транзитний сегмент 10.0.3.0/24, який імітує мережу Інтернет. Повна схема проходження трафіку наведена на рисунку 2.3.

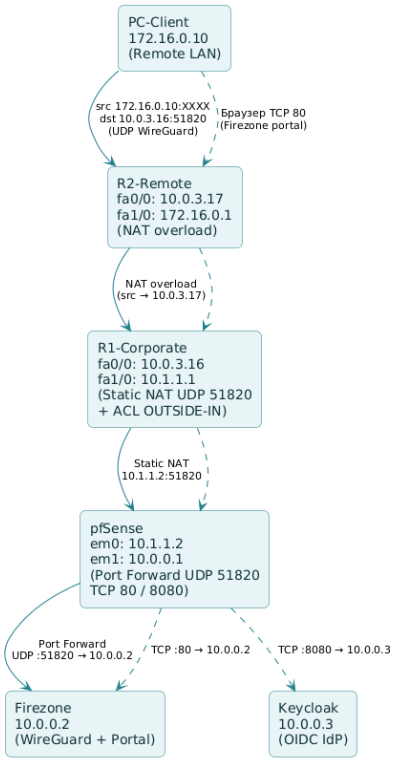


Рисунок 2.3 — Ланцюжок NAT та Port Forward

Вхідні WireGuard-пакети від зовнішніх клієнтів надходять із мережі Remote LAN (172.16.0.0/24) через R2-Remote, який виконує NAT overload і замінює адресу джерела на 10.0.3.17. Далі пакети потрапляють на R1-Corporate (10.0.3.16), де налаштовано статичний NAT: UDP-потік на порт 51820 перенаправляється на WAN-інтерфейс pfSense (10.1.1.2). Додатково на R1 налаштований ACL OUTSIDE-IN, що пропускає лише такі типи трафіку: DHCP (UDP 67/68), DNS (UDP 53), WireGuard (UDP 51820), встановлені TCP-з'єднання (established) та ICMP-відповіді. Весь інший трафік блокується з журналюванням (deny ip any any log), що реалізує принцип Zero Trust на периметрі.

pfSense, у свою чергу, виконує Port Forward для публікації трьох внутрішніх сервісів назовні, як показано у таблиці 2.5.

Таблиця 2.5

Правила Port Forwarding pfSense

Протокол	WAN-порт	Призначення	Сервіс
UDP	51820	10.0.0.2:51820	WireGuard (Firezone)
TCP	80	10.0.0.2:80	Firezone WebUI (для первинної OIDC-сесії)
TCP	8080	10.0.0.3:8080	Keycloak OIDC (обробка браузерних перенаправлень)

Порт TCP 80 (Firezone WebUI) публікується для того, щоб зовнішній клієнт міг ініціювати OIDC-автентифікацію через браузер до отримання WireGuard-конфігурації. Після встановлення тунелю цей порт стає необов'язковим — подальше управління відбувається всередині VPN. Keycloak (TCP 8080) публікується лише частково: браузер клієнта отримує redirect_uri від Firezone та звертається до Keycloak для передачі authorization_code. Адміністративна консоль Keycloak залишається доступною виключно через WireGuard-тунель.

Після розшифрування WireGuard-тунелю pfSense отримує пакети з IP-адресами джерела у підмережі 10.8.0.0/24 та застосовує firewall-правила для рольового розмежування доступу.

Правила фільтрації pfSense

pfSense виступає шлюзом між VPN-підмережею клієнтів (10.8.0.0/24) та серверним сегментом LAN (10.0.0.0/24). Правила організовані за принципом Default Deny: базове правило блокує весь трафік із VPN-підмережі, а дозвільні правила додаються вибірково для кожної ролі. Система підтримує дві ролі доступу — vpn-user та admin — відповідно до матриці, визначеної у таблиці 2.3 підрозділу 2.1. Проєкт правил міжмережевого екрана наведено у таблиці 2.6.

Таблиця 2.6

Проєкт правил міжмережевого екрана pfSense

№	Дія	Протокол	Джерело	Призначення	Порт dst	Опис
1	Pass	UDP	any	10.0.0.2	51820	WireGuard handshake
2	Pass	TCP	10.8.0.0/24	10.0.0.10	80	vpn-user → Internal Server
3	Pass	TCP	10.8.0.0/24	10.0.0.3	8080	vpn-user → Keycloak (лише вхід)
4	Block	any	10.8.0.0/24	10.0.0.1	any	Заборона доступу до pfSense WebGUI
5	Pass	any	admin IP/32	10.0.0.0/24	any	admin → повний доступ до LAN
6	Block	any	10.8.0.0/24	any	any	Default Deny — весь інший VPN-трафік

Правила застосовуються послідовно зверху вниз, тому порядок є критичним: дозвільні правила для конкретних ресурсів мають передувати загальному блокувальному правилу (правило №6). Логіку обробки пакета наведено на рисунку 2.4.

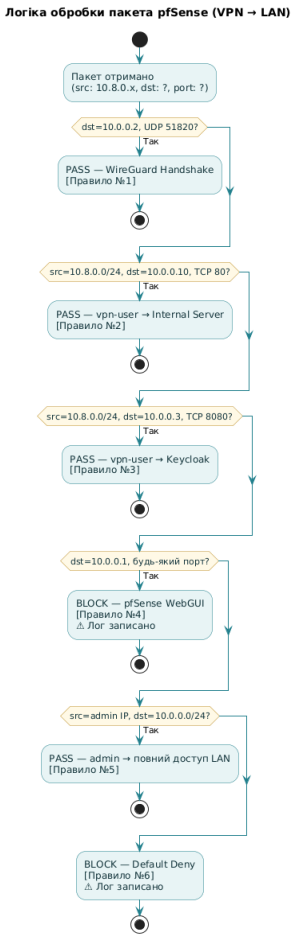


Рисунок 2.4 — Блок-схема обробки пакета pfSense

Журналювання увімкнено для правил блокування (правила №4 та №6), що фіксує всі спроби несанкціонованого доступу: часову мітку, IP-адресу джерела, IP та порт призначення. Ці журнали інтегруються із системою моніторингу, що описана у підрозділі 2.4.

Ключовим архітектурним аспектом є розміщення Firezone у сегменті LAN за pfSense, а не безпосередньо на WAN-інтерфейсі. pfSense пропускає на WAN лише зашифровані UDP-пакети WireGuard (порт 51820), тоді як розшифрований трафік клієнтів фільтрується на рівні LAN-інтерфейсу. Таким чином реалізується двоетапна перевірка: спочатку криптографічна автентифікація WireGuard — неможливо підключитися без валідного ключа, зареєстрованого у Firezone, —

потім рольова фільтрація pfSense — неможливо вийти за межі дозволеного навіть усередині тунелю. Ця схема є практичною реалізацією принципу Defense in Depth на мережевому рівні та унеможливлює як обхід тунелю (Bypass VPN), так і горизонтальне переміщення між сегментами (lateral movement). Детальний аналіз захисту від конкретних векторів атак наведено у підрозділі 2.5.

2.3 Розробка рольової моделі контролю доступу (RBAC) та варіанти використання системи

Рольова модель контролю доступу (Role-Based Access Control, RBAC) — це підхід до управління правами, при якому дозволи призначаються не безпосередньо користувачам, а ролям, а користувачі отримують права через призначення їм відповідних ролей. Стандарт RBAC формалізований у NIST INCITS 359-2004 та визначає чотири основні компоненти: користувачі (суб'єкти доступу), ролі (іменовані набори дозволів), дозволи (права на виконання операцій над ресурсами) та сесії (динамічна активація ролей). Ключова перевага RBAC перед прямим призначенням прав полягає в масштабованості: при зміні посадових обов'язків достатньо змінити роль користувача замість перегляду індивідуальних дозволів.

У межах дослідження проєктованої системи RBAC реалізується на двох рівнях: логічному — через Keycloak, де роль вбудовується у JWT Access Token як claim і використовується Firezone для авторизації підключення; та мережевому — через pfSense, де рольові обмеження реалізуються правилами фільтрації за IP-адресою клієнта у VPN-підмережі 10.8.0.0/24. Поєднання цих двох рівнів утворює цілісну систему контролю доступу, де ідентичність підтверджується криптографічно (Keycloak + WireGuard), а мережевий доступ обмежується на рівні брандмауера (pfSense).

У системі визначено дві ролі, що відображають типову ієрархію доступу в корпоративному середовищі. Характеристики кожної ролі наведено у таблиці 2.7.

Таблиця 2.7

Характеристики ролей системи

Атрибут	vpn-user	admin
Опис	Штатний співробітник / зовнішній клієнт	Системний адміністратор
Realm Role (Keycloak)	vpn-user	admin
Тунельна IP	динамічно /32	динамічно /32
Сесійний тайм-аут	8 годин	8 годин
Підтримка пристроїв	Linux, Windows, iOS, Android	Linux, Windows
Internal Server (10.0.0.10)	TCP 80	Повний (TCP 80, 443, 22)
Keycloak (10.0.0.3:8080)	Лише автентифікація	Admin Console
pfSense WebGUI (10.0.0.1)	Заблоковано	TCP 443
WAN / R1 (10.0.3.0/24)	Заблоковано	Заблоковано
Журналювання дій	Повне	Повне

Обидві ролі мають заблокований доступ до маршрутизатора R1 та WAN-сегменту — адміністрування мережевого обладнання виходить за межі VPN-доступу та виконується окремими засобами. Рольова модель побудована за принципом Least Privilege: кожен користувач отримує рівно той обсяг доступу, що необхідний для виконання його функцій.

Ролі визначаються у Keycloak Realm та призначаються обліковим записам через адміністративну консоль Keycloak. При підключенні клієнта Firezone отримує роль із JWT Access Token через userinfo-ендпойнт Keycloak та зіставляє її з правилами доступу. Оскільки Firezone призначає кожній ролі фіксований діапазон тунельних IP, pfSense однозначно ідентифікує роль за адресою джерела трафіку без необхідності аналізу вмісту пакетів. Ієрархію ролей наведено на рисунку 2.5.

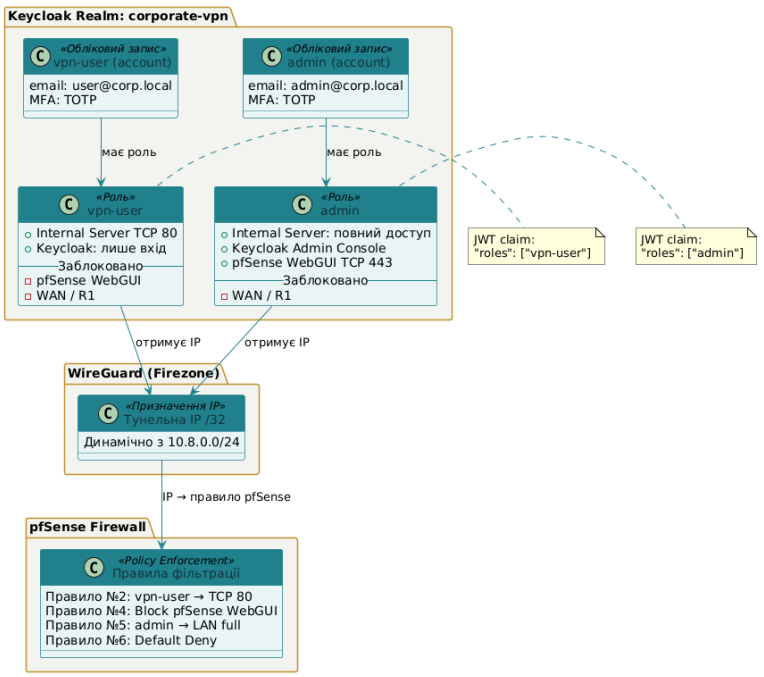


Рисунок 2.5 — Ієрархія ролей у Keycloak

Варіанти використання описують сценарії взаємодії користувачів із системою і є основою для тестування у розділі 3. Визначено двох основних користувачів: vpn-user та admin, а також системний користувач Keycloak IAM, що бере участь у кожному сценарії автентифікації. Загальну діаграму варіантів використання наведено на рисунку 2.6.

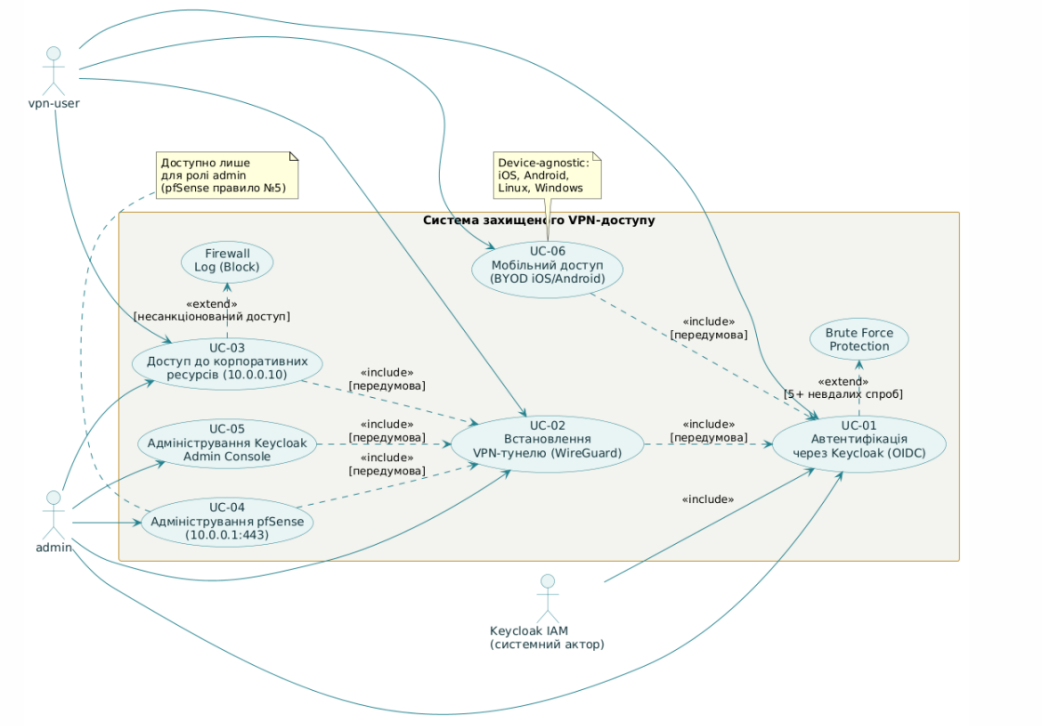


Рисунок 2.6 — Діаграма варіантів використання системи

UC-01 — Автентифікація через Keycloak (OIDC)

Користувачі: vpn-user, admin, Keycloak IAM

Основний потік: Користувач відкриває портал Firezone (http://10.0.0.2) → перенаправляється на сторінку автентифікації Keycloak (http://10.0.0.3:8080) → вводить логін та пароль → Keycloak верифікує облікові дані, визначає роль і генерує Access Token (JWT) → Firezone отримує токен, реєструє пристрій та видає конфігурацію WireGuard або QR-код для мобільного пристрою.

Альтернативний потік (Brute Force): Введено невірні дані 5 разів підряд → Keycloak фіксує події LOGIN_ERROR → спрацьовує Brute Force Protection: обліковий запис тимчасово блокується → подія журналюється з IP-адресою джерела.

UC-02 — Встановлення VPN-тунелю (WireGuard Handshake)

Передумова: Користувач успішно пройшов UC-01 та отримав конфігурацію WireGuard.

Основний потік: Клієнт активує WireGuard-інтерфейс (wg-quick up wg0 або Toggle у мобільному застосунку) → надсилається запит на криптографічний handshake на UDP 51820 → Firezone верифікує публічний ключ клієнта → тунель встановлюється, клієнту призначається тунельна IP у 10.8.0.0/24 → трафік до 10.0.0.0/24 починає маршрутизуватися через тунель.

Альтернативний потік (недійсний ключ): Клієнт із підробленим або недійсним приватним ключем надсилає handshake → WireGuard мовчки відхиляє пакет без відповіді (stealth mode) → тунель не встановлюється → доступ до 10.0.0.0/24 унеможливлений.

UC-03 — Доступ до корпоративних ресурсів

Передумова: VPN-тунель успішно встановлено (UC-02).

Основний потік (vpn-user): Користувач звертається до http://10.0.0.10 → трафік через тунель надходить на LAN-інтерфейс pfSense → pfSense перевіряє Source IP (10.8.0.x) → застосовує правило №2 таблиці 2.6 (Pass TCP → 10.0.0.10:80) → HTTP 200 OK.

Альтернативний потік (несанкціонований доступ): vpn-user намагається відкрити http://10.0.0.1 (pfSense WebGUI) → pfSense застосовує правило №4 таблиці 2.6 (Block) → з'єднання відхиляється → подія фіксується у Firewall Logs з поміткою «Block» та IP-адресою джерела.

UC-04, UC-05 — Адміністрування системи

Користувач: admin

Опис: Лише адміністратор має мережевий доступ до інтерфейсів управління pfSense (10.0.0.1:443) та Keycloak Admin Console (10.0.0.3:8080/admin) для управління користувачами, пристроями та перегляду журналів аудиту. Спроби доступу ролі vpn-user до цих адрес апаратно блокуються брандмауером pfSense відповідно до правила №4 таблиці 2.6.

UC-06 — Мобільний доступ (BYOD)

Користувач: vpn-user (мобільний пристрій iOS/Android)

Опис: Користувач встановлює застосунок WireGuard на мобільний пристрій та сканує QR-код із порталу Firezone → конфігурація імпортується автоматично → тунель встановлюється ідентично до UC-02. Архітектура системи є device-agnostic: логіка автентифікації та авторизації не залежить від типу пристрою чи операційної системи.

Розроблена рольова модель повністю відповідає принципу найменших привілеїв та забезпечує однозначний зв'язок між ідентичністю користувача (Keycloak), криптографічним ідентифікатором пристрою (WireGuard-ключ) та мережевими правами (pfSense). Варіанти використання UC-01–UC-06 є основою для тестових сценаріїв розділу 3.

2.4 Алгоритми взаємодії компонентів мережі та проєктування централізованої автентифікації

Проєктована система є розподіленою і складається з чотирьох активних компонентів: клієнтського вузла, Firezone (VPN-шлюзу на базі WireGuard), Keycloak та pfSense. Кожен компонент виконує строго визначену функцію, і жоден з них не може самостійно забезпечити повноцінний захист без взаємодії з іншими.

Ця взаємозалежність реалізує принцип Defense in Depth: зловмисник, що подолав один рівень захисту, неминуче стикається з наступним.

Взаємодію компонентів можна розділити на три функціональних алгоритми: алгоритм централізованої автентифікації та налаштування VPN-профілю, алгоритм встановлення захищеного WireGuard-з'єднання та алгоритм обробки й журналювання подій доступу. Кожен алгоритм описується окремо, після чого наводиться зведена діаграма повного циклу взаємодії.

Алгоритм 1 — Централізована автентифікація через Keycloak (OIDC Authorization Code Flow з PKCE)

Перший алгоритм описує процес ідентифікації користувача через Keycloak та отримання WireGuard-конфігурації через портал Firezone. Цей алгоритм є обов'язковою передумовою для встановлення VPN-тунелю, оскільки WireGuard-сервер приймає лише заздалегідь зареєстровані публічні ключі — саме Firezone, після успішної OIDC-автентифікації, забезпечує їх реєстрацію у конфігурацію WireGuard. Алгоритм наведено на рисунку 2.7.

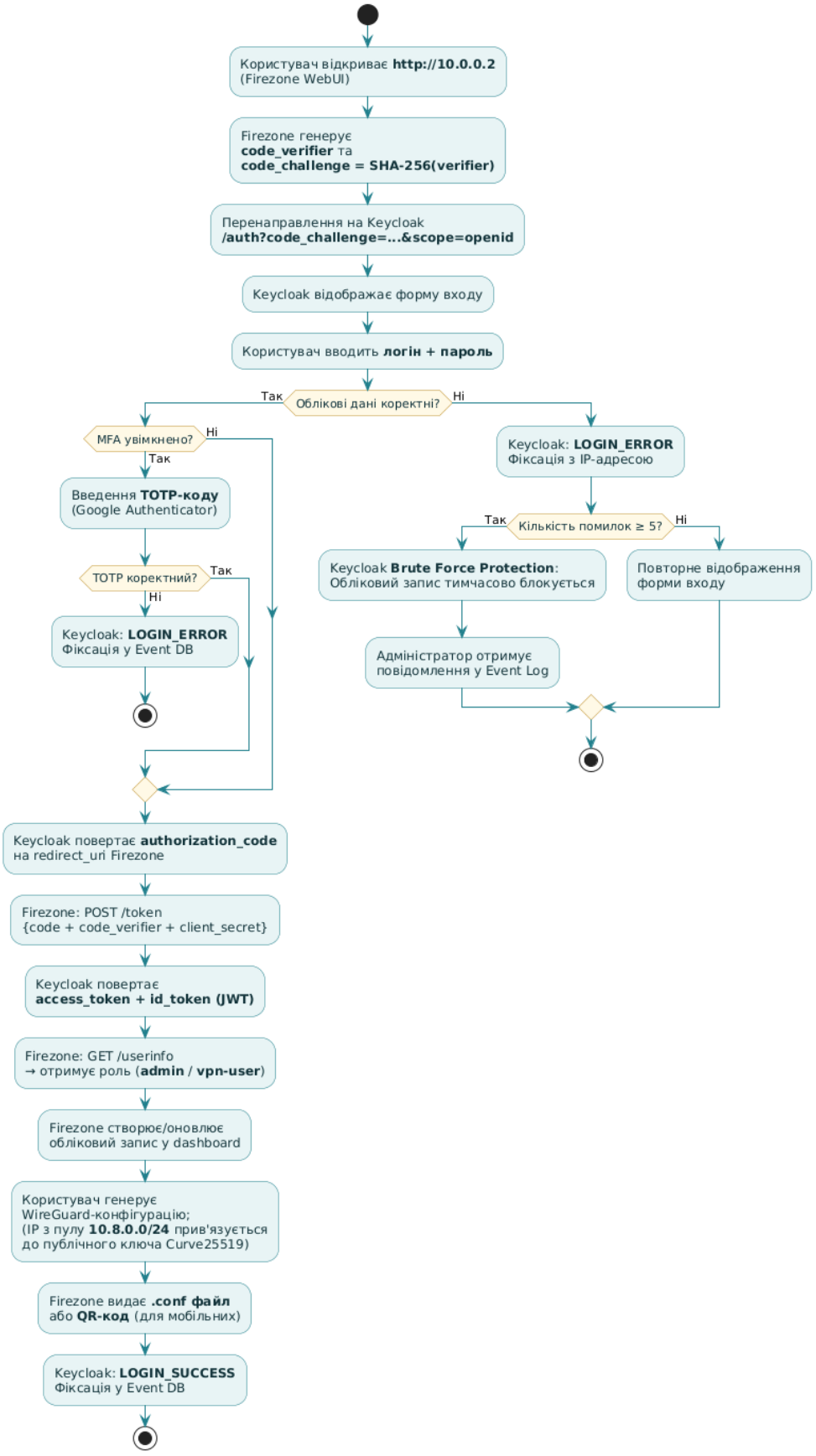


Рисунок 2.7 — Алгоритм централізованої автентифікації через Keycloak та налаштування VPN-профілю

Детальна послідовність кроків:

1. Користувач відкриває портал Firezone (http://10.0.0.2) та натискає «Sign in with Keycloak SSO»

2. Firezone генерує випадковий code_verifier та обчислює code_challenge = SHA-256(code_verifier), після чого перенаправляє браузер на authorization_endpoint Keycloak із параметрами response_type=code, code_challenge, scope=openid email profile

3. Keycloak (realm corporate-vpn) відображає сторінку входу — користувач вводить логін і пароль

4. Keycloak верифікує облікові дані та перенаправляє на redirect_uri Firezone із одноразовим code

5. Firezone обмінює code + code_verifier на access_token та id_token через token_endpoint

6. Firezone отримує дані користувача з userinfo_endpoint, включно з клієнтськими ролями (admin або vpn-user) завдяки налаштованому User Client Role Mapper

7. Firezone створює або оновлює обліковий запис та відкриває доступ до dashboard

8. Користувач генерує WireGuard-конфігурацію пристрою — Firezone статично прив'язує унікальну тунельну IP-адресу з пулу 10.8.0.0/24 до публічного ключа клієнта

Механізм PKCE є принципово важливим елементом безпеки: при кожному запиті Firezone генерує унікальний code_verifier, і перехоплений code без нього є марним для зловмисника. Відключення Direct Access Grants забороняє автентифікацію через пряму передачу логіна та пароля (ROPC flow), що відповідає вимогам сучасного стандарту OAuth 2.0.

Альтернативний потік (невдача автентифікації): Введено невірні дані → Keycloak фіксує подію LOGIN_ERROR → після визначеної кількості невдалих спроб спрацьовує політика Brute Force Protection: обліковий запис тимчасово блокується, подія журналюється з IP-адресою джерела.

Алгоритм 2 — Встановлення захищеного WireGuard-тунелю

Другий алгоритм описує криптографічне встановлення WireGuard-тунелю між клієнтом та Firezone-шлюзом. Цей алгоритм виконується після того, як користувач успішно пройшов OIDC-автентифікацію та завантажив конфігурацію пристрою (Алгоритм 1). Алгоритм наведено на рисунку 2.8.

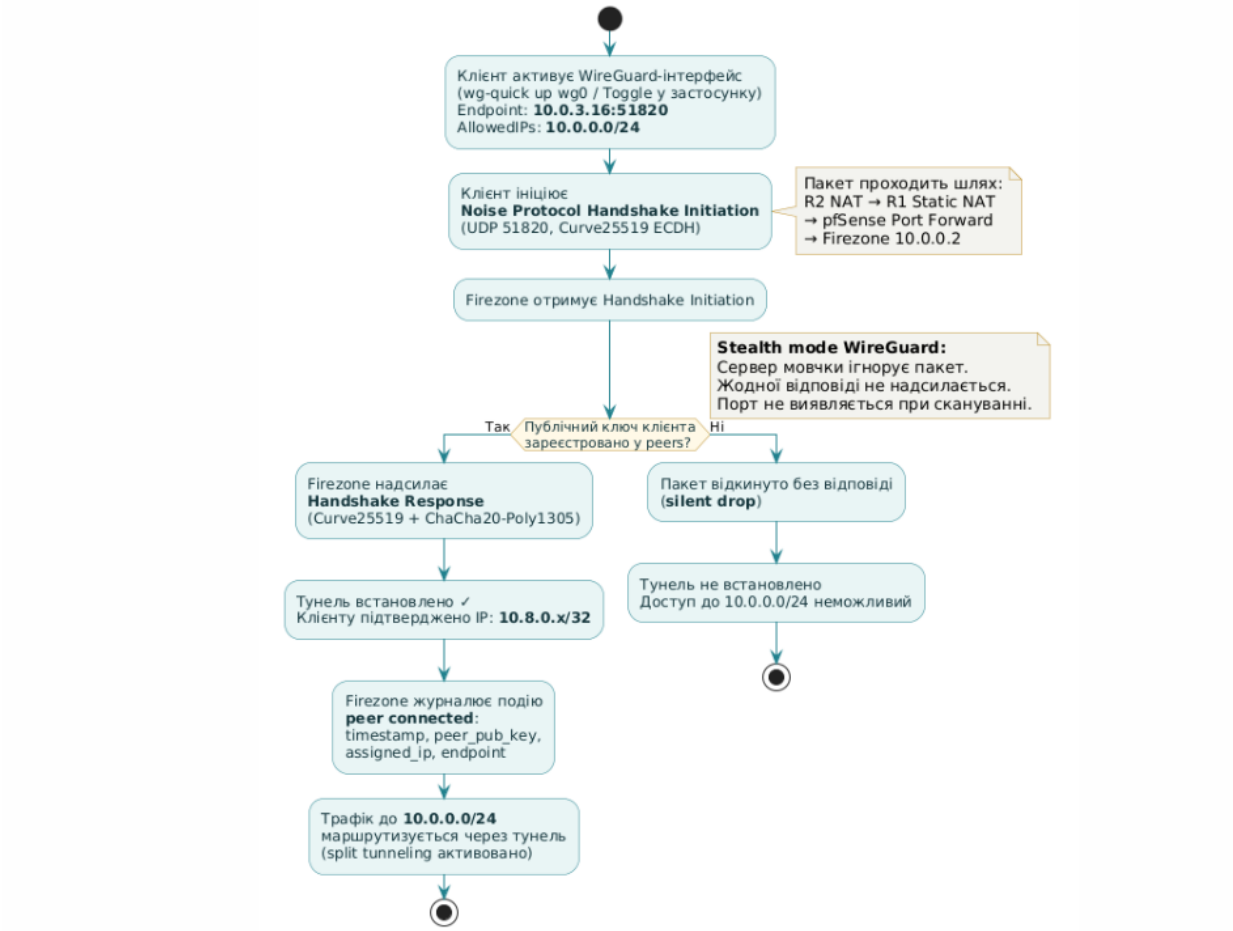


Рисунок 2.8 — Алгоритм встановлення WireGuard-тунелю (Noise Protocol Handshake)

Детальна послідовність кроків:

1. Клієнт активує WireGuard-інтерфейс із завантаженою конфігурацією (параметри: Endpoint = 10.0.3.16:51820, AllowedIPs = 10.0.0.0/24, PersistentKeepalive = 25)

2. Клієнт ініціює 1-RTT Noise Protocol Handshake до порту UDP 51820 Firezone-шлюзу (10.0.0.2)

3. Firezone-шлюз перевіряє публічний ключ клієнта у таблиці зареєстрованих peers — ключ має бути попередньо зареєстрований через портал (Алгоритм 1)

4. За умови збігу ключів handshake завершується успішно, встановлюється зашифрований L3-тунель; трафік до 10.0.0.0/24 починає маршрутизуватися через тунель (split tunneling)

5. Firezone фіксує подію peer connected у системному журналі з параметрами: timestamp, peer_pub_key, assigned_ip, endpoint

Принципово важливим є те, що WireGuard не надсилає жодного повідомлення про помилку у разі відхилення handshake — сервер просто ігнорує пакет. Ця властивість (stealth mode) унеможливлює сканування порту та виявлення Firezone-шлюзу засобами мережевої розвідки. Параметр AllowedIPs = 10.0.0.0/24 у конфігурації клієнта реалізує split tunneling — лише корпоративний трафік маршрутизується через тунель, решта трафіку залишається поза ним.

Алгоритм 3 — Обробка мережевого запиту та журналювання на pfSense

Третій алгоритм описує обробку трафіку клієнта після встановлення VPN-тунелю — від надходження пакета на pfSense до фіксації результату в журналі.

Детальна послідовність кроків:

1. Розшифрований клієнтський трафік надходить із WireGuard-тунелю на LAN-інтерфейс pfSense із Source IP з пулу 10.8.0.0/24

2. Оскільки Firezone статично прив'язує тунельну IP-адресу до публічного ключа клієнта (Cryptokey Routing), pfSense однозначно ідентифікує роль клієнта за Source IP без глибокої інспекції пакетів

3. pfSense послідовно перевіряє правила Firewall Rules LAN (таблиця 2.6) зверху вниз:

- Якщо Source IP належить адміністратору → правило №5: Pass any → 10.0.0.0/24
- Якщо Source IP належить vpn-user → правила №2–4: дозволено Internal Server (TCP 80) та Keycloak (TCP 8080); заблоковано pfSense WebGUI (10.0.0.1) з журналюванням

- Інший трафік → правило №6 Default Deny з журналюванням

4. Спрацювання правила блокування фіксується у Firewall Log із полями: timestamp, src_ip, dst_ip, dst_port, protocol, rule_id, action=BLOCK

Для наочного відображення взаємодії всіх компонентів у єдиному процесі наведено зведену діаграму послідовності повного циклу — від ініціації підключення до отримання відповіді від захищеного ресурсу (рисунок 2.9).

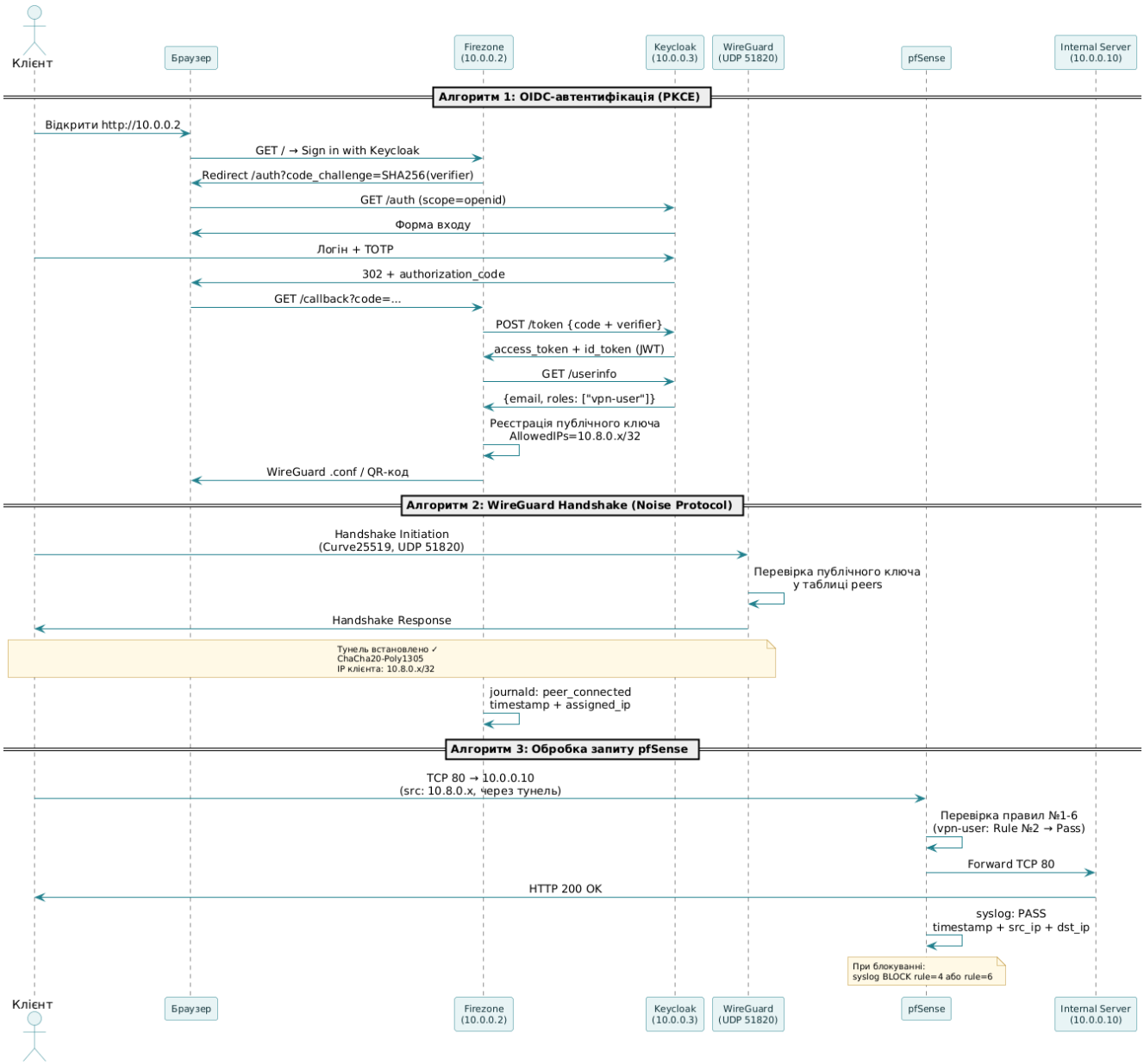


Рисунок 2.9 — Зведена діаграма взаємодії компонентів

Повноцінний аудит системи вимагає збору журналів з усіх трьох компонентів. Кожен компонент генерує події різного типу, що разом формують повний аудитний слід. Схему потоків журналювання наведено у таблиці 2.8.

Таблиця 2.8

Джерела та типи подій журналювання

Компонент	Тип події	Поля запису	Сховище
WireGuard (Firezone)	Peer connected / disconnected	timestamp, peer_pub_key, assigned_ip, endpoint	systemd journald
Keycloak	LOGIN_SUCCESS, LOGIN_ERROR, LOGOUT, TOKEN_ISSUED, USER_DISABLED	timestamp, user, ip, realm, event_type, error	Keycloak Event DB
pfSense	PASS / BLOCK	timestamp, src_ip, dst_ip, dst_port, protocol, rule_id, action	pfSense syslog

Усі три джерела журналів можуть бути налаштовані на передачу подій до централізованого syslog-сервера, що дозволяє кореляцію подій у часі: адміністратор може відстежити повний шлях інциденту — від встановлення WireGuard-тунелю (журнал Firezone) через OIDC-автентифікацію (журнал Keycloak) до конкретної мережевої дії (журнал pfSense). Принциповим є те, що IP-адреса клієнта у pfSense-логах однозначно відповідає конкретному обліковому запису Keycloak завдяки статичній прив'язці тунельної адреси в Firezone — це усуває проблему анонімності трафіку. Журналювання подій блокування на pfSense (правила №4 та №6 таблиці 2.6) є основою для негативних тест-кейсів підрозділу 3.3.

2.5 Аналіз загроз для спроєктованої системи

Аналіз загроз є невід'ємним етапом проєктування захищеної системи, оскільки дозволяє верифікувати архітектурні рішення шляхом їх зіставлення з реальними векторами атак. Для кожної загрози визначається її опис, модель зловмисника та механізми протидії, реалізовані в спроєктованій системі. Узагальнену матрицю загроз наведено у таблиці 2.9.

Таблиця 2.9

Матриця загроз та механізми протидії

ID	Загроза	Вектор атаки	Механізм протидії	Компонент
T1	Brute Force	Перебір паролів на порталі Firezone / Keycloak	Brute Force Detection, блокування акаунту після N невдалих спроб	Keycloak
T2	Stolen Credentials	Використання вкрадених логіна та пароля	Обов'язкова MFA (TOTP) + криптографічна ідентичність WireGuard	Keycloak + WireGuard
T3	Insider Threat	Горизонтальне переміщення у LAN з-під легітимного VPN	Default Deny + RBAC-правила pfSense	pfSense
T4	Bypass VPN	Прямий доступ до ресурсів LAN поза VPN-тунелем	Implicit Deny на WAN pfSense + ACL на R1	pfSense + Cisco IOS
T5	Network Scanning	Сканування інфраструктури (Nmap) із зовнішньої мережі	WireGuard Stealth Mode + ACL OUTSIDE-IN на R1	WireGuard + Cisco IOS

T1 — Brute Force

Опис загрози. Зловмисник намагається підібрати пароль облікового запису шляхом автоматизованого перебору через веб-портал Firezone або безпосередньо через authorization_endpoint Keycloak.

Модель зловмисника. Зовнішній атакуючий, що не має дійсних облікових даних, але знає або може вгадати ім'я користувача. Атака може проводитися як вручну, так і за допомогою інструментів автоматизованого перебору (Hydra, Burp Suite Intruder).

Механізм блокування в системі. Keycloak Realm corporate-vpn налаштовано з увімкненою політикою Brute Force Detection. Після фіксованої кількості невдалих спроб автентифікації (LOGIN_ERROR) обліковий запис тимчасово блокується — подальші спроби входу відхиляються до ручного розблокування адміністратором. Усі події LOGIN_ERROR фіксуються в Event Log Keycloak із зазначенням timestamp, user, ip, event_type, що дозволяє виявити атаку та встановити її джерело. Додатковим бар'єром є те, що навіть успішно підібраний пароль не дає доступу до VPN: для ролі admin MFA є обов'язковою, що унеможливлює завершення автентифікації без фізичного OTP-пристрою.

T2 — Stolen Credentials

Опис загрози. Зловмисник отримав дійсні логін і пароль легітимного користувача (наприклад, через фішинг або витік даних) та намагається використати їх для підключення до VPN.

Модель зловмисника. Атакуючий має коректні облікові дані (логін + пароль), але не має фізичного OTP-пристрою та WireGuard-конфігурації жертви — зокрема приватного ключа Curve25519.

Механізм блокування в системі. Система реалізує дворівневий захист, де кожен рівень є незалежним бар'єром.

Рівень 1 (Keycloak — MFA): Навіть з правильними логіном і паролем, без коректного TOTP-коду (що генерується застосунком Google Authenticator або аналогом) автентифікація не завершується. WireGuard-конфігурація не може бути завантажена без успішного входу в портал Firezone.

Рівень 2 (WireGuard — криптографічна ідентичність): Встановлення WireGuard-тунелю вимагає наявності приватного ключа клієнта на пристрої. Firezone реєструє публічний ключ при налаштуванні пристрою — handshake можливий лише з пристрою, де зберігається відповідний приватний ключ. Компрометація облікових даних без фізичного доступу до пристрою жертви не дозволяє встановити тунель. Фреймворк Noise Protocol забезпечує властивість forward secrecy: навіть у разі перехоплення сесійного трафіку та подальшого компрометування довгострокового ключа минулі сесії залишаються захищеними.

T3 — Insider Threat

Опис загрози. Легітимний користувач із роллю vpn-user, перебуваючи всередині VPN-тунелю, намагається отримати несанкціонований доступ до ресурсів, що виходять за межі його ролі — наприклад, до адміністративного інтерфейсу pfSense (10.0.0.1:443) або Keycloak Admin Console (10.0.0.3:8080/admin), або здійснює горизонтальне переміщення (lateral movement) у мережі LAN 10.0.0.0/24.

Модель зловмисника. Внутрішній зловмисник із дійсним VPN-з'єднанням та тунельною IP-адресою з пулу 10.8.0.0/24. Це може бути скомпрометований легітимний обліковий запис або навмисно зловмисний співробітник.

Механізм блокування в системі. Захист реалізується через систему правил pfSense (таблиця 2.6) на основі RBAC та принципу Default Deny. Оскільки Firezone статично прив'язує тунельну IP-адресу до облікового запису (Cryptokey Routing), pfSense однозначно ідентифікує роль за Source IP без глибокої інспекції пакетів:

- Правило №4 (таблиця 2.6): трафік із IP vpn-user до 10.0.0.1 (pfSense WebGUI) → BLOCK із журналюванням
- Правило №6 (таблиця 2.6): будь-який трафік із 10.8.0.0/24, що не потрапив під попередні дозвільні правила → Default Deny із журналюванням

Спроби несанкціонованого доступу фіксуються у Firewall Log pfSense із зазначенням src_ip, dst_ip, dst_port, action=BLOCK, що є підставою для виявлення підозрілої активності.

T4 — Bypass VPN

Опис загрози. Зловмисник або легітимний клієнт намагається отримати доступ до внутрішніх ресурсів LAN (10.0.0.0/24) безпосередньо із зовнішньої мережі, не встановлюючи WireGuard-тунель — наприклад, шляхом прямого HTTP-запиту або ICMP-пінгу до вузла 10.0.0.10.

Модель зловмисника. Будь-який вузол із адресою WAN-сегменту (172.16.0.0/24 або зовнішньої мережі), що намагається досягти LAN-ресурсів поза VPN.

Механізм блокування в системі. Реалізовано дворівневий периметровий захист.

Рівень 1 (Cisco IOS ACL на R1): На зовнішньому інтерфейсі застосовано розширений ACL OUTSIDE-IN з кінцевим правилом deny ip any any log. ACL дозволяє лише UDP 51820 (WireGuard handshake) та необхідні службові протоколи (DHCP, DNS, ICMP reply), блокуючи весь інший вхідний трафік до досягнення pfSense.

Рівень 2 (pfSense WAN implicit deny): pfSense не має жодних дозвільних правил для вхідного трафіку на WAN-інтерфейсі, окрім Port Forward порту 51820/UDP до Firezone (таблиця 2.5). Будь-який пакет, що минув ACL і досяг pfSense, відхиляється правилом implicit deny на WAN. Ресурси LAN (10.0.0.0/24) фізично недосяжні без проходження WireGuard-тунелю.

T5 — Network Scanning

Опис загрози. Зловмисник проводить розвідку мережевої інфраструктури за допомогою засобів сканування портів (Nmap, Masscan) з метою виявлення відкритих портів, сервісів та вразливостей на публічно доступних адресах системи.

Модель зловмисника. Зовнішній атакуючий на етапі первинної розвідки (Reconnaissance у моделі MITRE ATT&CK), що не має жодних дійсних облікових даних.

Механізм блокування в системі. WireGuard реалізує концепцію Stealth Mode (Silent Drop): сервер не відповідає жодним чином на пакети, що не відповідають дійсному криптографічному handshake (Noise Protocol). Навіть якщо зловмисник знає порт 51820/UDP, він не отримає жодної відповіді — ні ICMP port unreachable, ні RST-пакета, без жодного підтвердження існування сервісу. З точки зору мережевого сканера порт виглядає ідентично відфільтрованому порту на мовчазному хості. Ця властивість структурно відповідає принципу Single Packet Authorization (SPA) в архітектурі SDP, розглянутій у підрозділі 1.2. Додатково, ACL OUTSIDE-IN на маршрутизаторі R1 блокує TCP SYN-сканування та інші типи зондів ще на периметрі мережі (deny ip any any log). У результаті сканування Nmap із параметрами -sS -sU -p 51820 повертає статус filtered або взагалі не виявляє жодного відгуку від цільового вузла.

Проведений аналіз підтверджує, що спроєктована система реалізує принцип Defense in Depth: жодна із розглянутих загроз не може бути реалізована шляхом подолання лише одного рівня захисту. Keycloak забезпечує захист рівня ідентичності (T1, T2), WireGuard — захист транспортного рівня (T2, T5), pfSense — захист мережевого рівня (T3, T4), а Cisco IOS ACL — захист зовнішнього периметру (T4, T5). Ця багаторівнева архітектура є практичною реалізацією вимог,

сформульованих у підрозділі 1.4, та слугує основою для розробки тест-кейсів у підрозділі 3.3.

Висновок до другого розділу

У рамках проєктування розроблено трирівневу архітектуру захищеного віддаленого доступу, що реалізує принципи Zero Trust та Defense in Depth: периметровий захист через pfSense, захищений транспорт на базі WireGuard/Firezone та централізована ідентифікація через Keycloak. Ключовим архітектурним рішенням є статична прив'язка тунельної IP-адреси до облікового запису користувача (Cryptokey Routing у Firezone), що дозволяє pfSense однозначно ідентифікувати роль клієнта за адресою джерела трафіку та застосовувати гранульовані правила RBAC без глибокої інспекції пакетів.

Проведений аналіз загроз підтвердив архітектурну стійкість системи до п'яти ключових векторів атак: brute force та компрометація облікових даних нейтралізуються комбінацією Brute Force Detection у Keycloak і обов'язковою MFA; горизонтальне переміщення у LAN унеможливлюється правилами Default Deny на pfSense; спроби обходу VPN блокуються дворівневим периметровим захистом (ACL Cisco IOS + implicit deny pfSense); мережева розвідка унеможливлюється завдяки Stealth Mode WireGuard, який структурно відтворює принцип SPA архітектури SDP. Спроєктована система та варіанти використання UC-01–UC-06 є вихідними даними для практичної реалізації та тестування у Розділі 3.

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ ЗАХИЩЕНОГО ДОСТУПУ

3.1 Розгортання інфраструктури та базове налаштування мережевого шлюзу

Розгортання лабораторного стенду виконувалось у середовищі GNS3 з інтеграцією віртуальних машин Oracle VirtualBox. Стенд відтворює реалістичну корпоративну мережу з двома географічно розподіленими вузлами, централізованим міжмережевим екраном та інфраструктурою автентифікації на основі принципів Zero Trust. Загальна топологія стенду наведена на рисунку 3.1.

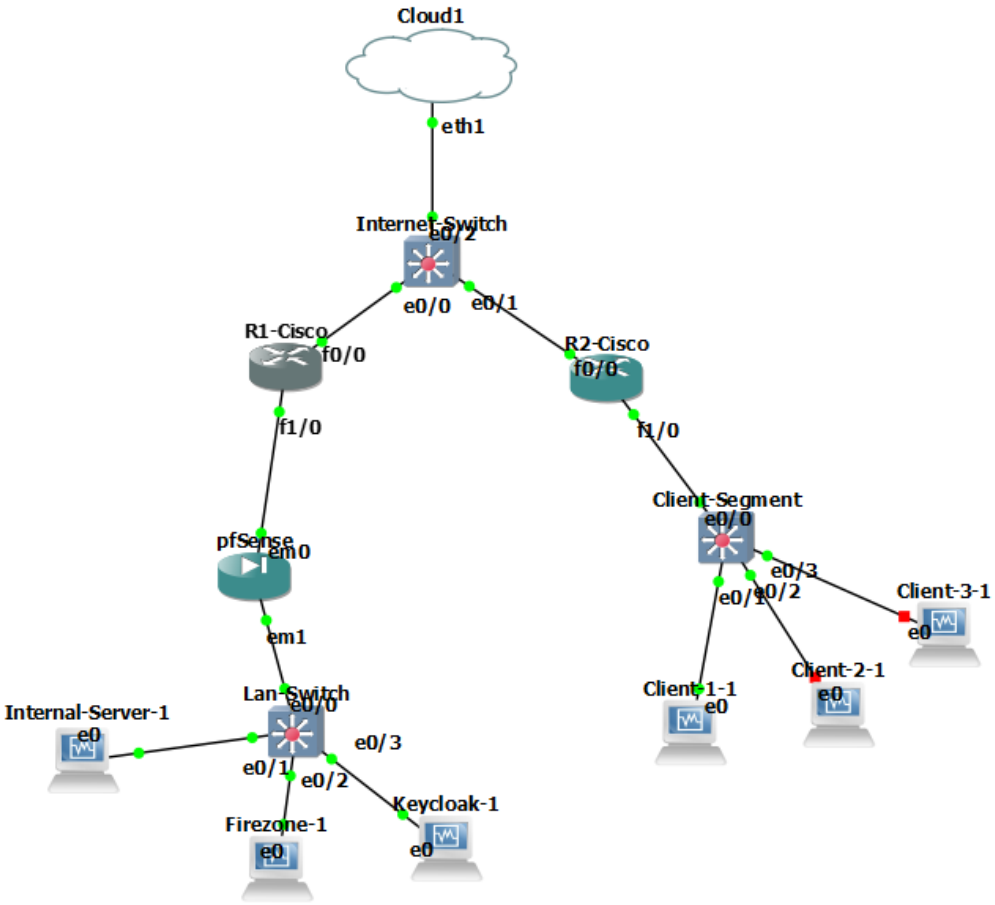


Рисунок 3.1 — Топологія мережевого стенду в середовищі GNS3

Мережа розподілена на чотири логічні зони безпеки відповідно до принципу мікросегментації Zero Trust.

Зона імітації Інтернету побудована на базі комутатора Internet-SW, до якого підключені обидва граничних маршрутизатори R1 та R2 через хмарний вузол NAT

Cloud GNS3. Цей вузол забезпечує підключення до реальної мережі через NAT-механізм гіпервізора VirtualBox, що дозволяє завантажувати пакети та перевіряти зовнішню зв'язність.

Корпоративна захищена зона (мережа 10.0.0.0/24) розміщена за міжмережевим екраном pfSense і містить такі компоненти: pfSense (10.0.0.1) як міжмережевий екран і шлюз, Firezone (10.0.0.2) як WireGuard VPN-сервер, Keycloak (10.0.0.3) як сервер ідентифікації та авторизації, Internal Server (10.0.0.10) як внутрішній корпоративний веб-ресурс. Адресний план мережі наведено у таблиці 3.1.

Таблиця 3.1

Адресний план мережевого стенду

Вузол	Інтерфейс	IP-адреса	Призначення
R1-Corporate	Fa0/0	10.0.3.16/24 (DHCP)	WAN до Internet-SW
R1-Corporate	Fa1/0	10.1.1.1/30	Transit до pfSense
pfSense	em0 (WAN)	10.1.1.2/30	Підключення до R1
pfSense	em1 (LAN)	10.0.0.1/24	Корпоративна мережа
Firezone	eth0	10.0.0.2/24	VPN-сервер
Keycloak	eth0	10.0.0.3/24	SSO-сервер
Internal Server	eth0	10.0.0.10/24	Внутрішній ресурс
R2-Remote	Fa0/0	10.0.3.17/24 (DHCP)	WAN до Internet-SW
R2-Remote	Fa1/0	172.16.0.1/24	LAN клієнтської мережі
PC-Client	enp0s3	172.16.0.10/24 (DHCP)	Тестовий клієнт
WireGuard пул	wg0	10.8.0.0/24	VPN клієнти

Віддалена клієнтська мережа (172.16.0.0/24) знаходиться за маршрутизатором R2-Remote та імітує мережу філії або віддалених користувачів. R2 надає клієнтам адреси через вбудований DHCP-сервер та забезпечує вихід до Інтернету через NAT overload.

R1-Corporate виконує роль граничного маршрутизатора корпоративної зони. Конфігурація інтерфейсів наведена на рисунку 3.2.

```
R1-Corporate#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            10.0.3.16       YES DHCP   up                    up
FastEthernet1/0            10.1.1.1        YES NVRAM  up                    up
FastEthernet1/1            unassigned      YES NVRAM  administratively down down
NVI0                       unassigned      YES unset  administratively down down
```

Рисунок 3.2 — Виведення команди show ip interface brief на R1-Corporate

Центральним елементом конфігурації R1 є статична NAT-трансляція для WireGuard-трафіку та захисний ACL на WAN-інтерфейсі. Повна конфігурація NAT та маршрутизації:

ip nat inside source static udp 10.1.1.2 51820 interface FastEthernet0/0 51820

ip nat inside source list 10 interface FastEthernet0/0 overload

ip route 0.0.0.0 0.0.0.0 10.0.3.2

ip route 172.16.0.0 255.255.255.0 10.1.1.2

access-list 10 permit 10.1.1.0 0.0.0.3

Статична NAT-трансляція (static udp) перенаправляє вхідний UDP-трафік на порт 51820 безпосередньо на pfSense WAN-інтерфейс (10.1.1.2). Статичний маршрут до 172.16.0.0/24 через pfSense (10.1.1.2) забезпечує симетричну маршрутизацію відповідей WireGuard назад до клієнтів R2, що є критичною умовою для підтримки UDP-асоціацій. Список доступу 10 навмисно обмежено лише підмережею /30 між R1 та pfSense — розширення NAT на інші підмережі призводило б до некоректної трансляції зворотних пакетів Firezone.

На вхідному WAN-інтерфейсі FastEthernet0/0 застосовано розширений ACL OUTSIDE-IN, що реалізує принцип мінімальних привілеїв Zero Trust на рівні периметру мережі. ACL дозволяє лише такий вхідний трафік: DHCP (UDP 67→68), DNS (UDP 53), встановлені TCP-сесії (established), вхідний UDP 51820 для WireGuard та ICMP echo-reply для діагностики. Весь інший трафік журналюється та відкидається, що підтверджується у системному журналі R1 записами виду %SEC-6-IPACCESSLOGP: list OUTSIDE-IN denied. Конфігурація ACL наведена на рисунку 3.3.

```
R1-Corporate#show running-config | section ip access-list extended OUTSIDE-IN
ip access-list extended OUTSIDE-IN
 permit udp any eq bootps any eq bootpc
 permit udp any eq domain any
 permit tcp any any established
 permit udp any any eq 51820
 permit icmp any any echo-reply
 permit icmp any any time-exceeded
 permit icmp any any unreachable
 deny   ip any any log
```

Рисунок 3.3 — Конфігурація ACL OUTSIDE-IN на R1-Corporate

Верифікацію роботи NAT-трансляцій виконано командою show ip nat translations. На рисунку 3.4 відображено активну статичну UDP-асоціацію для порту 51820 та динамічні записи для вихідного трафіку pfSense.

```
R1-Corporate#show ip nat tra
Pro Inside global      Inside local       Outside local      Outside global
udp 10.0.3.16:123      10.1.1.2:123       162.159.200.1:123  162.159.200.1:123
udp 10.0.3.16:123      10.1.1.2:123       193.84.22.254:123  193.84.22.254:123
udp 10.0.3.16:123      10.1.1.2:123       217.12.206.12:123  217.12.206.12:123
tcp 10.0.3.16:7002     10.1.1.2:7002      91.189.91.98:80    91.189.91.98:80
udp 10.0.3.16:12061    10.1.1.2:12061     8.8.8.8:53         8.8.8.8:53
tcp 10.0.3.16:13070    10.1.1.2:13070     162.159.61.4:443   162.159.61.4:443
tcp 10.0.3.16:14060    10.1.1.2:14060     34.107.243.93:443  34.107.243.93:443
tcp 10.0.3.16:17656    10.1.1.2:17656     34.107.243.93:443  34.107.243.93:443
tcp 10.0.3.16:23517    10.1.1.2:23517     172.64.41.4:443    172.64.41.4:443
tcp 10.0.3.16:36462    10.1.1.2:36462     34.107.243.93:443  34.107.243.93:443
udp 10.0.3.16:51820    10.1.1.2:51820     10.0.3.17:53652    10.0.3.17:53652
udp 10.0.3.16:51820    10.1.1.2:51820     ---                ---
tcp 10.0.3.16:54772    10.1.1.2:54772     162.159.61.4:443   162.159.61.4:443
```

Рисунок 3.4 — Таблиця NAT-трансляцій R1-Corporate під час активного WireGuard-сеансу

Налаштування маршрутизатора R2-Remote

R2-Remote імітує маршрутизатор мережі філії. DHCP-сервер видає адреси клієнтам у діапазоні 172.16.0.10–172.16.0.254, шлюзом є 172.16.0.1. Вихід до Інтернету забезпечується через NAT overload на інтерфейсі FastEthernet0/0.

Критичним аспектом, виявленим у ході налаштування, стала необхідність правильного порядку правил в ACL OUTSIDE-IN. Оскільки Cisco IOS обробляє ACL послідовно зверху вниз і зупиняється на першому збігу, правило дозволу вхідного UDP 51820 обов'язково має передувати кінцевому правилу deny ip any any. Розміщення дозвільного правила після deny призводило до блокування всіх відповідей WireGuard-сервера, що фіксувалось у журналі R2 записами %SEC-6-IPACCESSLOGP: list OUTSIDE-IN denied udp 10.0.3.16(51820). Кінцева коректна конфігурація ACL:

```
ip access-list extended OUTSIDE-IN
 permit udp any eq bootps any eq bootpc
 permit udp any eq domain any
 permit udp any eq 51820 any
 permit tcp any any established
 permit icmp any any echo-reply
```

permit icmp any any time-exceeded

permit icmp any any unreachable

deny ip any any log

Налаштування міжмережевого екрану pfSense

pfSense розгорнуто у вигляді віртуальної машини VirtualBox та підключено до GNS3 топології через два мережевих адаптери: WAN (em0, 10.1.1.2/30) підключений до R1, LAN (em1, 10.0.0.1/24) підключений до корпоративного комутатора LAN-SW. Початкове налаштування виконувалось через веб-інтерфейс pfSense за адресою http://10.0.0.1.

Налаштування Port Forward виконано у розділі Firewall → NAT → Port Forward. Три правила забезпечують перенаправлення вхідних з'єднань до відповідних серверів корпоративної зони. Конфігурація правил Port Forward наведена на рисунку 3.5.

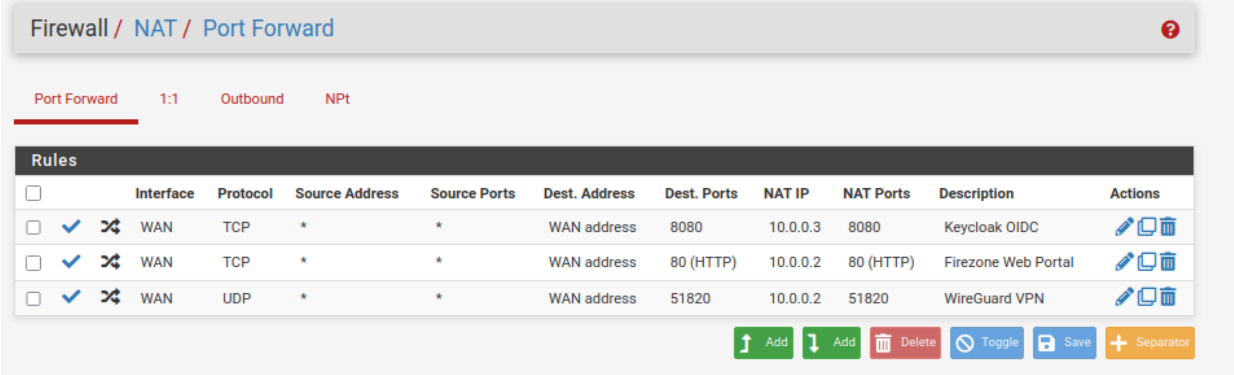


Рисунок 3.5 — Правила Port Forward у pfSense для WireGuard, Firezone та Keycloak

Автоматично створені pfSense відповідні WAN firewall rules виявились недостатніми — через Default Deny policy додатково створено явне Pass-правило для UDP 51820 з будь-якого джерела. Також для коректної роботи в лабораторному середовищі, де всі адреси є приватними (RFC 1918), на WAN-інтерфейсі деактивовано параметр Block private networks у розділі Interfaces → WAN. Налаштування WAN firewall rules наведено на рисунку 3.6.

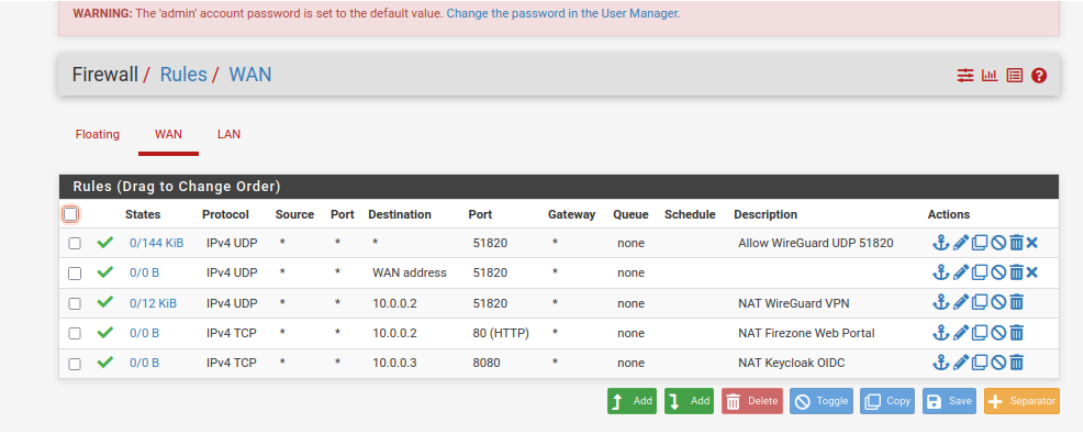


Рисунок 3.6 — Правила Firewall Rules WAN у pfSense

Налаштування Outbound NAT виконано у режимі Hybrid (Firewall → NAT → Outbound). Додатково створено правило з параметром Static Port для джерела 10.0.0.2/32 на порту UDP 51820 (рисунок 3.7). Цей параметр забороняє pfSense змінювати вихідний порт при NAT-трансляції відповідей WireGuard-сервера, що є обов'язковою умовою збереження UDP-асоціацій — без Static Port pfSense випадково змінював порт 51820 на інший, і маршрутизатор R1 не міг зіставити відповідь зі статичним NAT-записом.

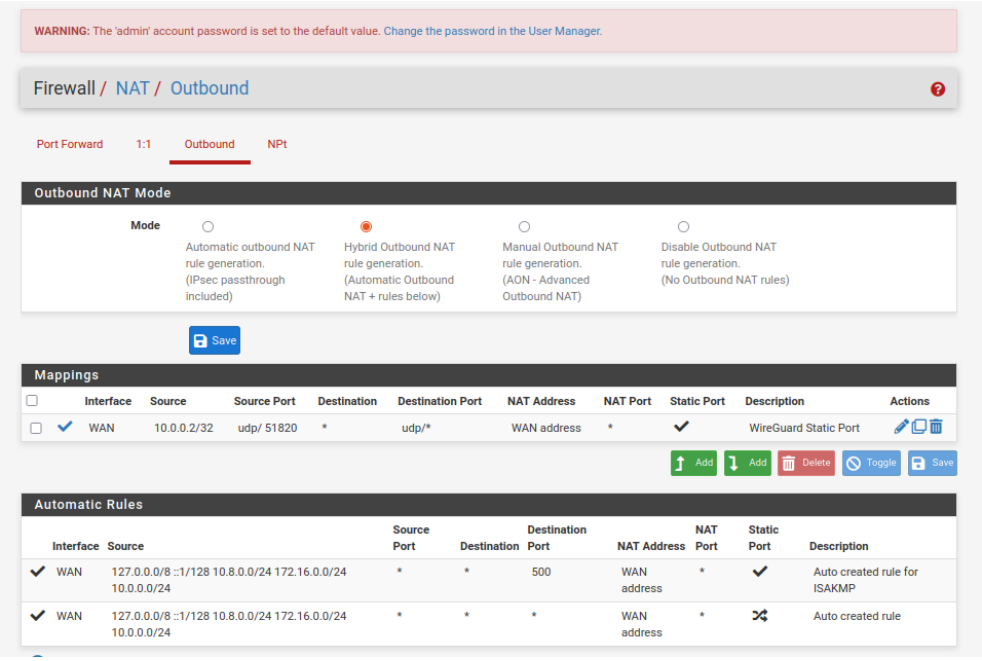


Рисунок 3.7 — Правило Outbound NAT зі Static Port для WireGuard у pfSense

Налаштування статичних маршрутів виконано у розділі System → Routing → Static Routes. Маршрут мережі 10.8.0.0/24 через шлюз 10.0.0.2 (Firezone) забезпечує маршрутизацію трафіку до WireGuard-клієнтів з корпоративної мережі. Таблиця маршрутизації pfSense наведена на рисунку 3.8.

IPv4 Routes

Destination	Gateway	Flags	Uses	MTU	Interface	Expire
default	10.1.1.1	UGS	6	1500	em0	
10.0.0.0/24	link#2	U	1	1500	em1	
10.0.0.1	link#8	UHS	3	16384	lo0	
10.1.1.0/30	link#1	U	4	1500	em0	
10.1.1.2	link#8	UHS	5	16384	lo0	
10.8.0.0/24	10.0.0.2	UGS	7	1500	em1	
127.0.0.1	link#8	UH	2	16384	lo0	
172.16.0.0/24	10.1.1.1	UGS	8	1500	em0	

Рисунок 3.8 — Таблиця маршрутизації pfSense (Diagnostics → Routes)

Таким чином, в результаті налаштування мережевої інфраструктури забезпечено коректну маршрутизацію та трансляцію адрес у всіх сегментах стенду. Конфігурація граничних маршрутизаторів R1 та R2, міжмережевого екрану pfSense та правил Port Forward створює необхідну мережеву основу для подальшого розгортання компонентів Zero Trust архітектури. Налаштований ланцюжок PC-Client - R2 - Internet-SW - R1 - pfSense - Firezone забезпечує транспортний рівень для WireGuard-тунелів, автентифікація та авторизація в яких реалізується засобами Keycloak та Firezone, що описано у підрозділі 3.2.

3.2 Налаштування сервера автентифікації Keycloak та його інтеграція з VPN-сервісом

Централізована автентифікація є ключовим елементом сучасної корпоративної VPN-інфраструктури, оскільки дозволяє управляти доступом користувачів з єдиної точки, застосовувати єдині політики безпеки та реалізовувати механізм Single Sign-On (SSO). У даній роботі роль Identity and Access Management (IAM) системи виконує сервер Keycloak, інтегрований з VPN-сервісом Firezone за протоколом OpenID Connect (OIDC).

Розгортання та початкове налаштування Keycloak:

Keycloak розгорнуто у вигляді Docker-контейнера на окремій віртуальній машині з адресою 10.0.0.3. Сервіс доступний за адресою http://10.0.0.3:8080. Для ізоляції конфігурації корпоративного VPN від інших потенційних застосунків у Keycloak створено окремий realm — логічно відокремлений простір ідентифікації зі власними користувачами, клієнтами та політиками.

Realm створено у розділі Keycloak Admin Console → Create Realm з такими параметрами:

Realm name: corporate-vpn

Display name: Corporate VPN

Enabled: ON

У налаштуваннях realm (Realm settings → Login) визначено політики входу відповідно до корпоративних вимог (рисунок 3.9):

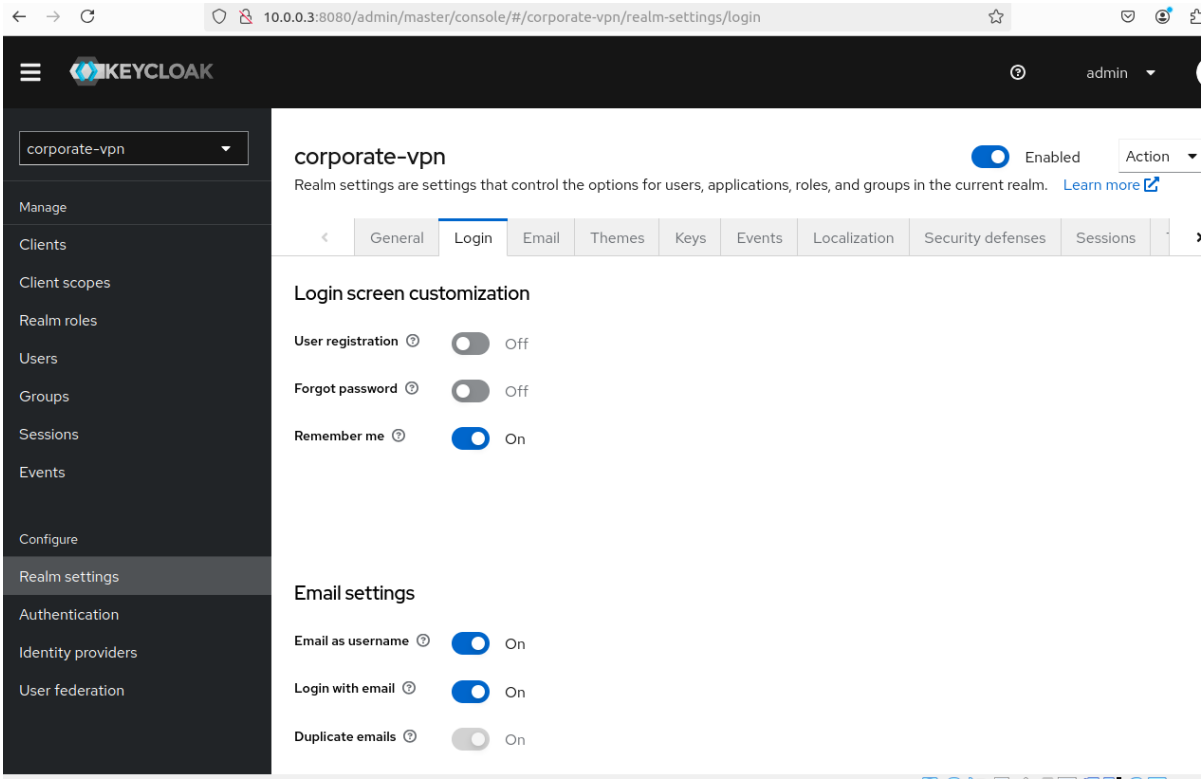


Рисунок 3.9 — Налаштування Login політик realm corporate-vpn у Keycloak

У розділі Realm settings → Tokens встановлено часові обмеження сесій відповідно до корпоративної політики безпеки:

Access Token Lifespan: 15 хвилин

SSO Session Idle: 30 хвилин

SSO Session Max: 10 годин

Створення та налаштування OIDC-клієнта Firezone

OIDC-клієнт є представником застосунку (Firezone) у системі Keycloak і визначає умови, за яких застосунок може запитувати автентифікацію користувачів. Клієнт створено у розділі Clients → Create client з такими базовими параметрами (рисунок 3.10):

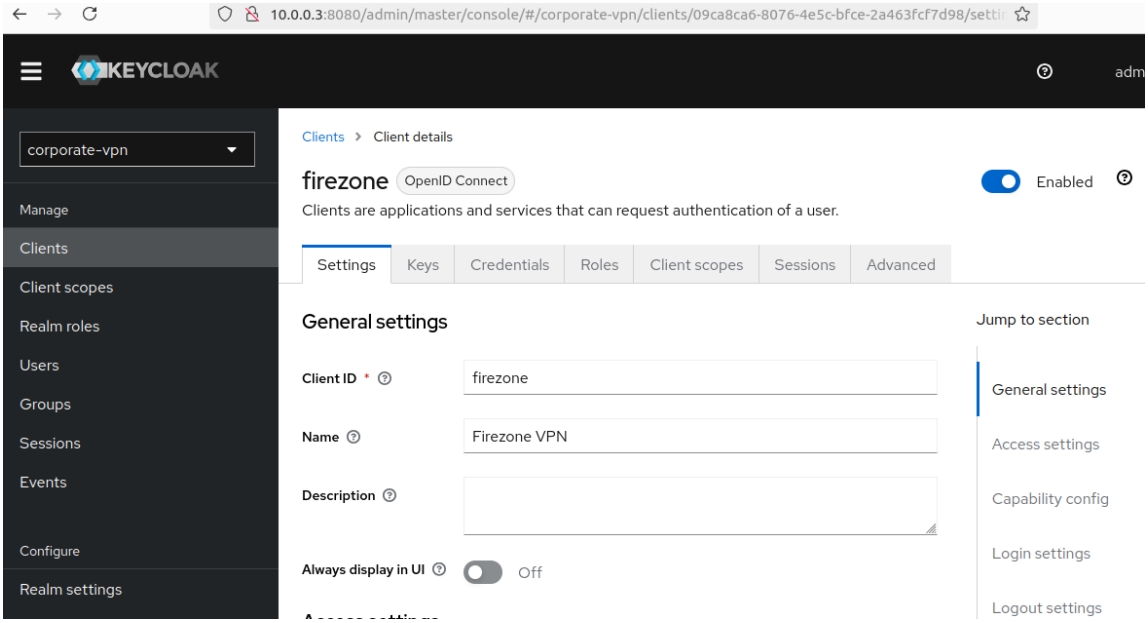


Рисунок 3.10 — Налаштування OIDC-клієнта Firezone у Keycloak

Вибір типу confidential client з увімкненою Client authentication означає що Firezone при обміні authorization code на токен зобов'язаний додатково пред'явити Client Secret — це унеможливлює підробку запитів від імені клієнта. Відключення Direct access grants забороняє автентифікацію через пряму передачу логіну та паролю (Resource Owner Password Credentials flow), що є вимогою сучасних стандартів безпеки OAuth 2.0.

У розділі Login settings клієнта визначено дозволені URI перенаправлення (рисунок 3.11):

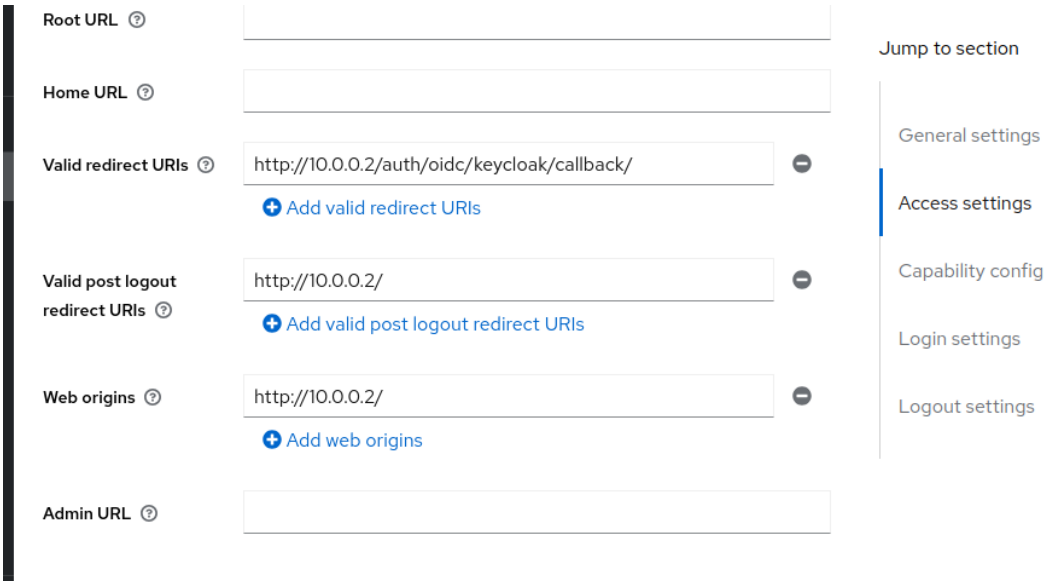


Рисунок 3.11 — Налаштування redirect URI для клієнта Firezone

Точне співпадіння redirect_uri є критичною вимогою безпеки OIDC — у процесі налаштування виявлено що Firezone додає завершальний символ / до callback URI (/auth/oidc/keycloak/callback/), тому у Valid redirect URIs прописано URI з цим символом.

У розділі Advanced → Advanced Settings активовано підтримку PKCE (Proof Key for Code Exchange) з методом S256 (рисунок 3.12):

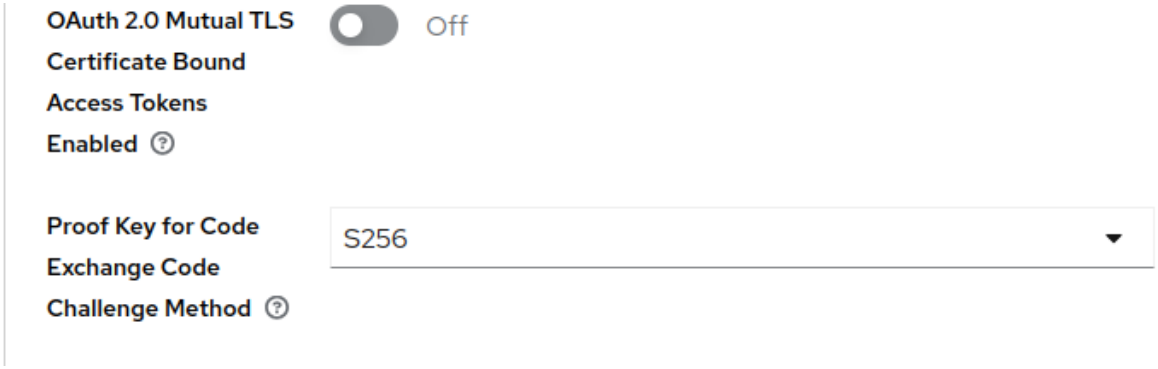


Рисунок 3.12 — Активація PKCE S256 для клієнта Firezone

PKCE є розширенням протоколу OAuth 2.0, що захищає від атак перехоплення authorization code. При кожному запиті автентифікації Firezone генерує випадковий code_verifier, обчислює його SHA-256 хеш (code_challenge) та передає його до Keycloak. При обміні code на токен Firezone надсилає оригінальний code_verifier, який Keycloak верифікує — таким чином перехоплений code без code_verifier не може бути використаний зловмисником.

Налаштування ролей та mapper токенів

Рольова модель доступу реалізована через Client Roles у Keycloak. У розділі Clients → Firezone → Roles створено дві ролі (рисунок 3.13):

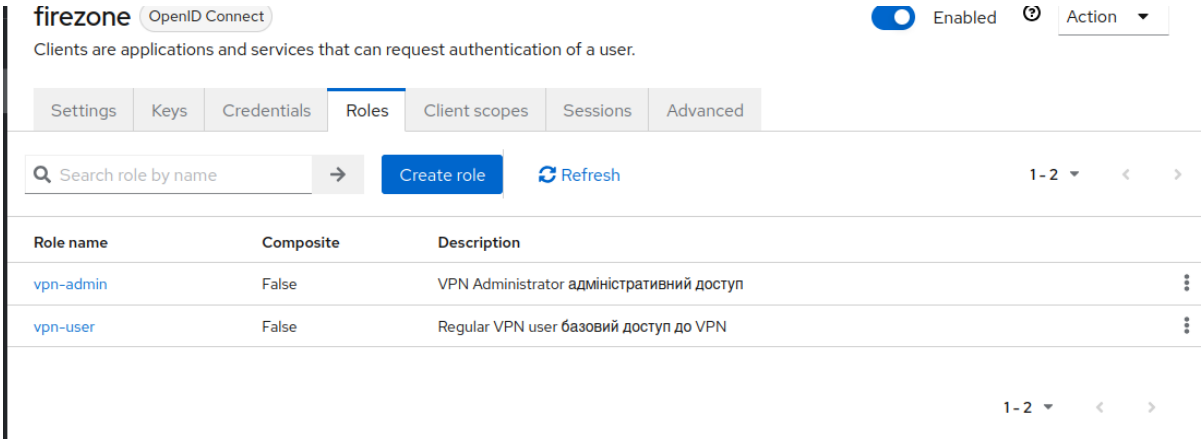


Рисунок 3.13 — Client Roles клієнта Firezone у Keycloak

Для передачі ролей користувача у складі OIDC-токенів до Firezone налаштовано Mapper у розділі Clients → Firezone → Client scopes → Firezone-dedicated → Mappers → Add mapper → User Client Role (рисунок 3.14):

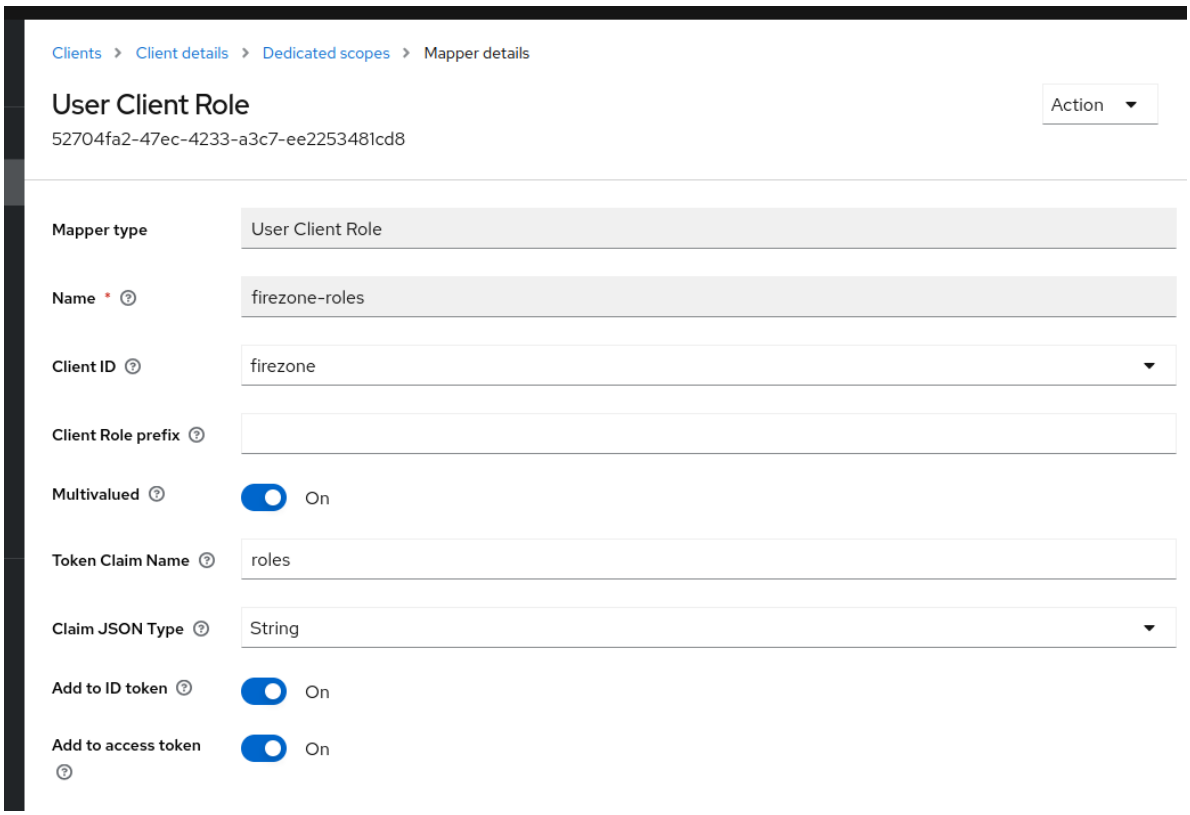


Рисунок 3.14 — Налаштування Mapper для передачі ролей у токені

Без цього mapper Keycloak не включає клієнтські ролі до JWT-токенів, і Firezone не отримує інформацію про рольову приналежність користувача. Параметр Multivalued: ON дозволяє передавати одночасно декілька ролей одного користувача.

Створення облікових записів користувачів

Відповідно до рольової моделі створено два тестових облікових записи. Адміністратор VPN та звичайний користувач VPN.

Після створення у розділі Role mappings → Assign role → Filter by clients → firezone призначено обидві ролі: vpn-admin та vpn-user, а звичайному користувачу призначено роль vpn-user.

Обліковий запис користувачів у Keycloak наведено на рисунку 3.15.

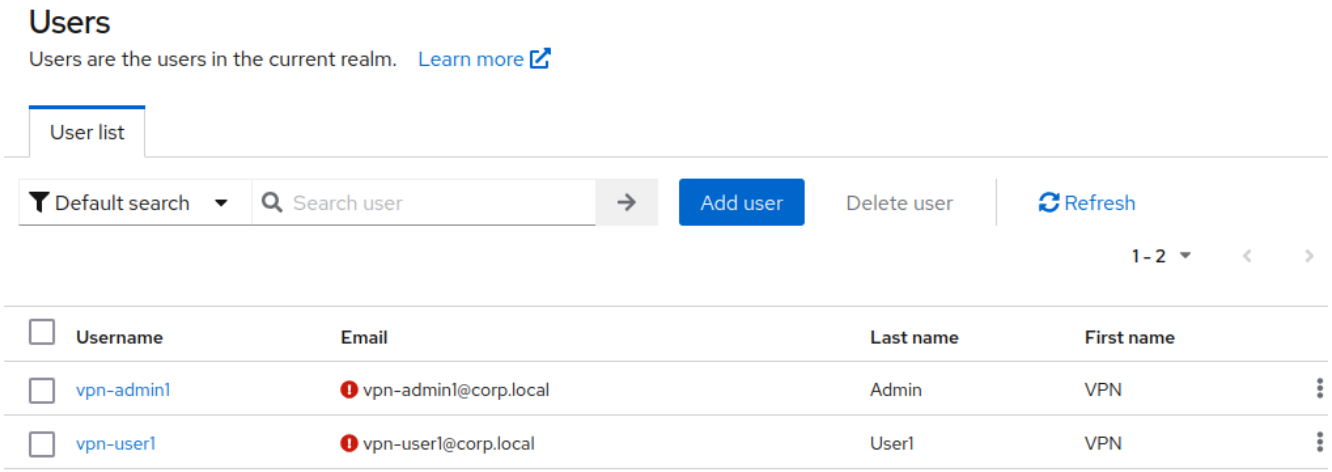


Рисунок 3.15 — Список користувачів realm corporate-vpn

Налаштування Firezone та інтеграція з Keycloak OIDC

Firezone розгорнуто у вигляді набору Docker-контейнерів на VM з адресою 10.0.0.2. Конфігурацію виконано через файл .env у директорії ~/.firezone/:

EXTERNAL_URL=http://10.0.0.2

DEFAULT_ADMIN_EMAIL=admin@corp.local

DEFAULT_ADMIN_PASSWORD=O9Y9qIdvNj5g+5aY

SECURE_COOKIES=false

WIREGUARD_IPV4_NETWORK=10.8.0.0/24

WIREGUARD_IPV4_ADDRESS=10.8.0.1

WIREGUARD_PORT=51820

DNS=8.8.8.8,8.8.4.4

Параметр SECURE_COOKIES=false необхідний для лабораторного середовища без HTTPS — у продакшен-середовищі використовується TLS та значення true.

Підключення Keycloak як OIDC-провайдера виконано у веб-інтерфейсі Firezone у розділі Settings → Security → Single Sign-On → Add OpenID Connect Provider (рисунок 3.16):

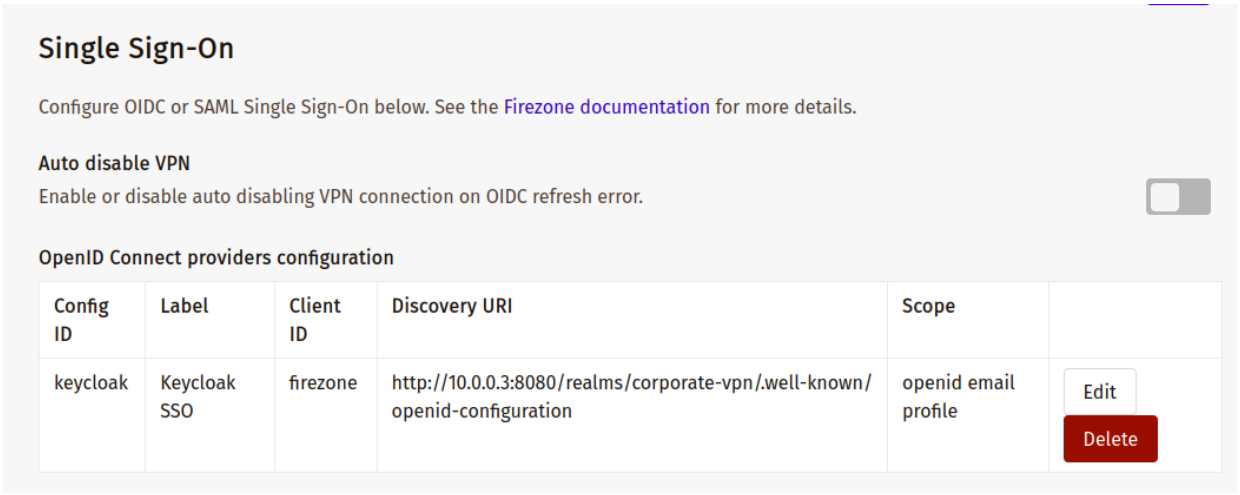


Рисунок 3.16 — Налаштування OIDC-провайдера Keycloak у Firezone

Discovery Document URI є стандартним OIDC endpoint, за яким Firezone автоматично отримує всі необхідні параметри Keycloak: authorization_endpoint, token_endpoint, userinfo_endpoint, jwks_uri та інші. Це спрощує конфігурацію та забезпечує автоматичне оновлення параметрів при зміні конфігурації Keycloak.

Після інтеграції сторінка входу Firezone (http://10.0.0.2) відображає кнопку "Sign in with Keycloak SSO" поряд зі стандартною формою. Процес автентифікації реалізує повний Authorization Code Flow з PKCE (рисунок 3.17):

1. Користувач натискає Sign in with Keycloak SSO у Firezone
2. Firezone генерує code_verifier та code_challenge (SHA-256), формує запит до Keycloak authorization_endpoint з параметрами response_type=code, code_challenge, scope=openid email profile
3. Keycloak відображає сторінку входу — користувач вводить email та пароль
4. Keycloak верифікує облікові дані та перенаправляє на redirect_uri Firezone з параметром code
5. Firezone обмінює code + code_verifier на access_token та id_token через token_endpoint
6. Firezone отримує дані користувача з userinfo_endpoint, включно з клієнтськими ролями завдяки налаштованому Mapper
7. Firezone створює або оновлює обліковий запис та відкриває доступ до dashboard

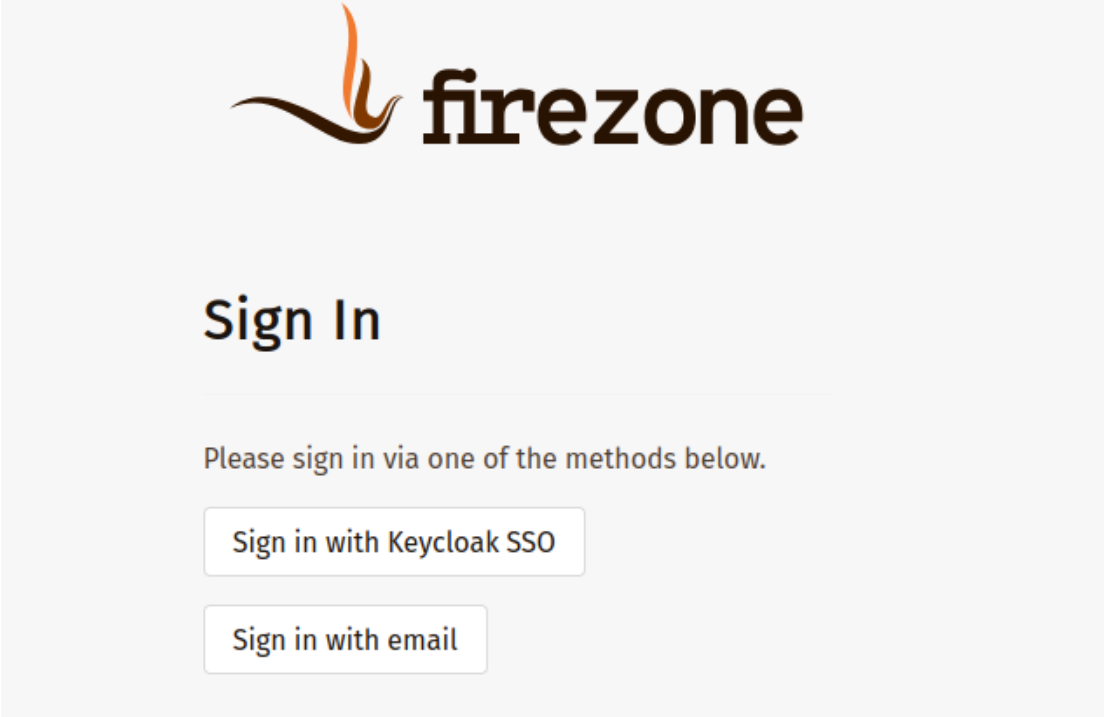


Рисунок 3.17 — Сторінка входу Firezone з кнопкою Keycloak SSO

Успішна автентифікація підтверджена входом облікового запису vpn-admin1@corp.local — Firezone відображає персоналізований dashboard з активною VPN-сесією терміном до 24 годин та можливістю генерації WireGuard-конфігурацій для пристроїв.

Після автентифікації через Keycloak SSO користувач отримує доступ до розділу керування пристроями у Firezone. Для кожного підключеного пристрою система генерує унікальну WireGuard-конфігурацію з індивідуальною парою ключів та тунельною адресою з пулу 10.8.0.0/24. Приклад згенерованої конфігурації для тестового клієнта:

[Interface]

PrivateKey = y9u70ro9sU56PMl6sFgin9BXYfRctB2g5sAXGufJE2M=

Address = 10.8.0.236/32, fd00::3:2:c6/128

MTU = 1280

DNS = 8.8.8.8, 8.8.4.4

[Peer]

PresharedKey = G5oM4yDB7jQV/JnX1eoCXUO+qduEakhpGFPuRssnX4c=

PublicKey = TlJTvfPkNkJga2hE43If0FvFwf5nAREz5i/53oqgow8=

AllowedIPs = 10.0.0.0/24

Endpoint = 10.0.3.16:51820

PersistentKeepalive = 25

Параметр AllowedIPs = 10.0.0.0/24 реалізує split tunneling — через VPN-тунель маршрутизується лише трафік до корпоративної мережі, решта трафіку користувача залишається поза тунелем. Це знижує навантаження на VPN-сервер та відповідає принципу мінімально необхідного доступу. Інтерфейс керування пристроями Firezone наведено на рисунку 3.18.

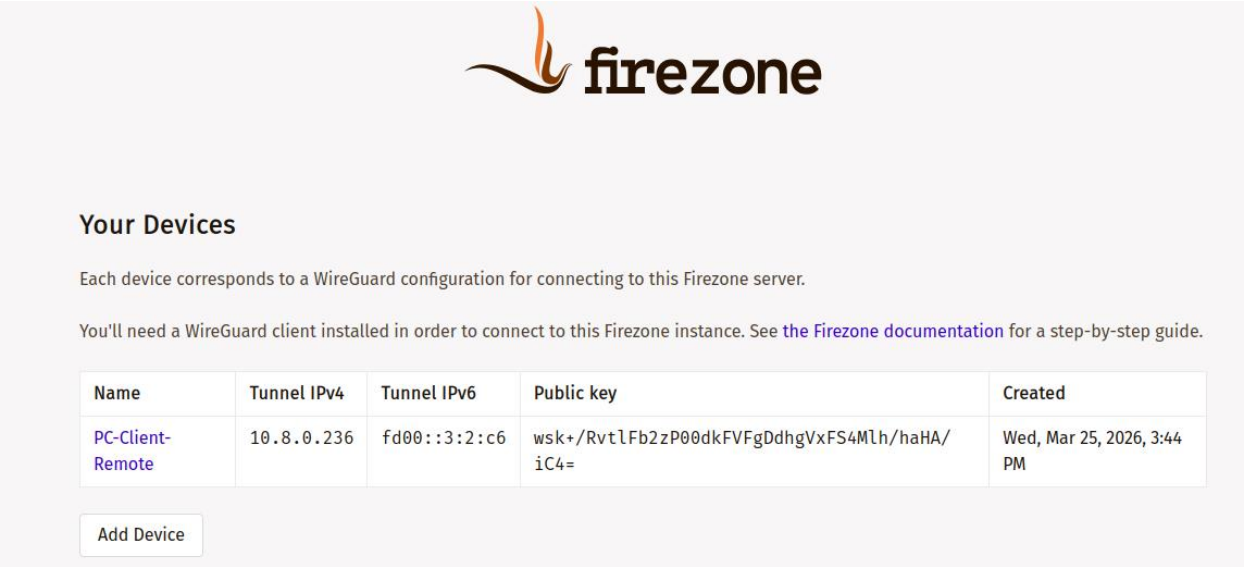


Рисунок 3.18 — Інтерфейс керування пристроями у Firezone після автентифікації через Keycloak

Таким чином, інтеграція Keycloak та Firezone за протоколом OIDC з Authorization Code Flow та PKCE забезпечує централізовану автентифікацію користувачів VPN-інфраструктури з підтримкою рольової моделі доступу, що відповідає вимогам сучасних IAM-технологій та є основою для реалізації гнучких політик доступу, описаних у підрозділі 3.3.

3.3 Налаштування політик журналювання подій та опис інтерфейсу керування доступом

Журналювання подій та централізоване керування доступом є невід'ємними складовими корпоративної VPN-інфраструктури, оскільки забезпечують аудит дій користувачів, фіксацію спроб несанкціонованого доступу та оперативне реагування на інциденти безпеки. У реалізованій інфраструктурі журналювання

охоплює три рівні: мережевий рівень (pfSense), рівень VPN-сервісу (Firezone) та рівень ідентифікації (Keycloak).

Налаштування правил міжмережевого екрану pfSense з журналюванням

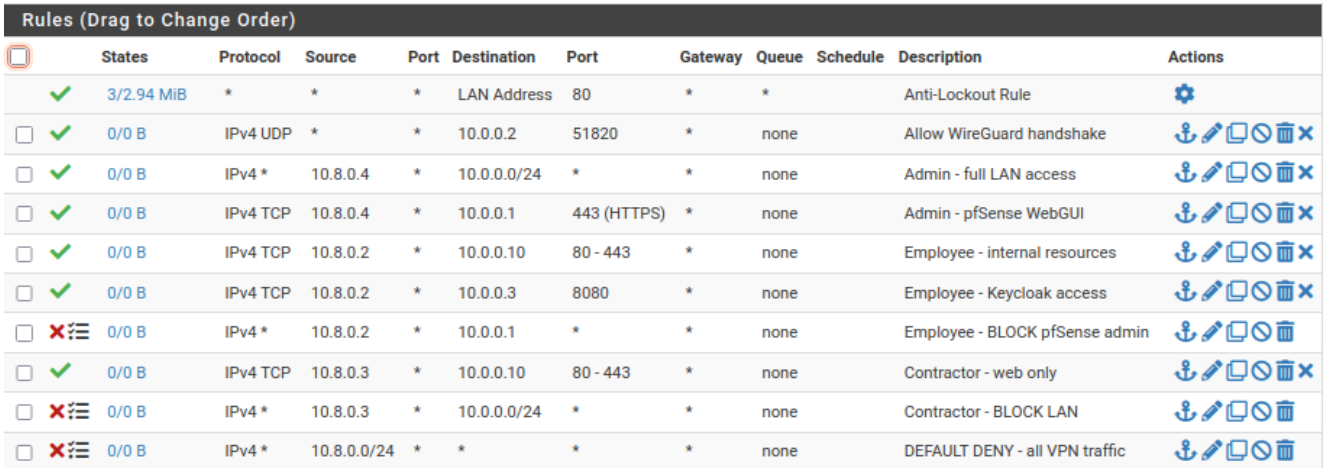
Політика доступу на рівні мережі реалізована через набір правил у розділі Firewall → Rules → LAN міжмережевого екрану pfSense. Правила впорядковано відповідно до принципу "дозволяти мінімально необхідне, блокувати все інше" та наведено у таблиці 3.2.

Оскільки архітектура протоколу WireGuard базується на маршрутизації за криптографічними ключами (Cryptokey Routing) і не використовує динамічну видачу адрес (DHCP), VPN-сервер Firezone під час генерації профілю пристрою статично прив'язує унікальну тунельну IP-адресу (наприклад, 10.8.0.2/32) до публічного ключа клієнта. Це гарантує незмінність IP-адреси пристрою та дозволяє реалізувати надійні рольові політики керування доступом на міжмережевому екрані pfSense.

Таблиця 3.2

Правила Firewall Rules LAN у pfSense

№	Дія	Протокол	Джерело	Призначення	Порт	Журнал	Опис
1	Pass	UDP	any	10.0.0.2	51820	—	Allow WireGuard handshake
2	Pass	any	10.8.0.4/32	10.0.0.0/24	any	—	Admin — full LAN access
3	Pass	TCP	10.8.0.2/32	10.0.0.10	80, 443, 22	—	Employee — internal resources
4	Pass	TCP	10.8.0.2/32	10.0.0.3	8080	—	Employee — Keycloak access
5	Block	any	10.8.0.2/32	10.0.0.1	any	+	Employee — BLOCK pfSense admin
6	Pass	TCP	10.8.0.3/32	10.0.0.10	80, 443	—	Contractor — web only
7	Block	any	10.8.0.3/32	10.0.0.0/24	any	+	Contractor — BLOCK LAN
8	Block	any	10.8.0.0/24	any	any	+	DEFAULT DENY — all VPN traffic



Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓	3/2.94 MiB	*	*	*	LAN Address	80	*	*		Anti-Lockout Rule	⚙
✓	0/0 B	IPv4 UDP	*	*	10.0.0.2	51820	*	none		Allow WireGuard handshake	
✓	0/0 B	IPv4 *	10.8.0.4	*	10.0.0.0/24	*	*	none		Admin - full LAN access	
✓	0/0 B	IPv4 TCP	10.8.0.4	*	10.0.0.1	443 (HTTPS)	*	none		Admin - pfSense WebGUI	
✓	0/0 B	IPv4 TCP	10.8.0.2	*	10.0.0.10	80 - 443	*	none		Employee - internal resources	
✓	0/0 B	IPv4 TCP	10.8.0.2	*	10.0.0.3	8080	*	none		Employee - Keycloak access	
✗	0/0 B	IPv4 *	10.8.0.2	*	10.0.0.1	*	*	none		Employee - BLOCK pfSense admin	
✓	0/0 B	IPv4 TCP	10.8.0.3	*	10.0.0.10	80 - 443	*	none		Contractor - web only	
✗	0/0 B	IPv4 *	10.8.0.3	*	10.0.0.0/24	*	*	none		Contractor - BLOCK LAN	
✗	0/0 B	IPv4 *	10.8.0.0/24	*	*	*	*	none		DEFAULT DENY - all VPN traffic	

Рисунок 3.19 — Правила Firewall Rules LAN у pfSense

Правила журналювання (позначені +) активовано на всіх Block-правилах, що забезпечує фіксацію кожної спроби несанкціонованого доступу. Зокрема:

- Правило 5 фіксує спроби звичайних VPN-користувачів (10.8.0.2) отримати доступ до адміністративного інтерфейсу pfSense (10.0.0.1) — такий доступ заборонено за принципом розмежування привілеїв
- Правило 7 фіксує спроби підрядників (10.8.0.3) звернутись до будь-яких ресурсів корпоративної мережі, окрім веб-порталу
- Правило 8 є фінальним Default Deny для всього WireGuard-пулу (10.8.0.0/24) і фіксує будь-який трафік, що не потрапив під жодне з дозвільних правил

Порядок правил є критичним — pfSense обробляє їх зверху вниз і зупиняється на першому збігу, тому дозвільні правила розміщено вище відповідних блокуючих.

Перегляд журналів подій pfSense

Журнали мережевих подій доступні у розділі Status → System Logs → Firewall. Інтерфейс дозволяє фільтрувати події за інтерфейсом, протоколом, IP-адресою джерела або призначення та портом. Приклад запису журналу при спробі доступу без VPN-тунелю:

Mar 25 14:37:52 WAN Block 10.0.3.17:60880 → 10.0.0.2:51820 UDP

Mar 25 14:38:03 WAN Block 10.0.3.17:60880 → 10.0.0.2:51820 UDP

Ці записи відповідають ситуації коли PC-Client намагався встановити WireGuard-з'єднання до того як було виправлено правила доступу на WAN-інтерфейсі. Після коректного налаштування такі записи зникли з журналу, що підтвердило правомірність трафіку. Інтерфейс журналів pfSense наведено на рисунку 3.20.



Рисунок 3.20 — Журнал подій Firewall у pfSense

Для фільтрації записів за конкретним портом або адресою використовується вбудований пошук. Наприклад, фільтр за портом 51820 дозволяє відстежити всю активність WireGuard-тунелів: ініціювання з'єднань, спроби підключення з неавторизованих адрес та розриви сесій.

Журналювання подій у Keycloak

Keycloak забезпечує детальне журналювання подій автентифікації та адміністративних дій.

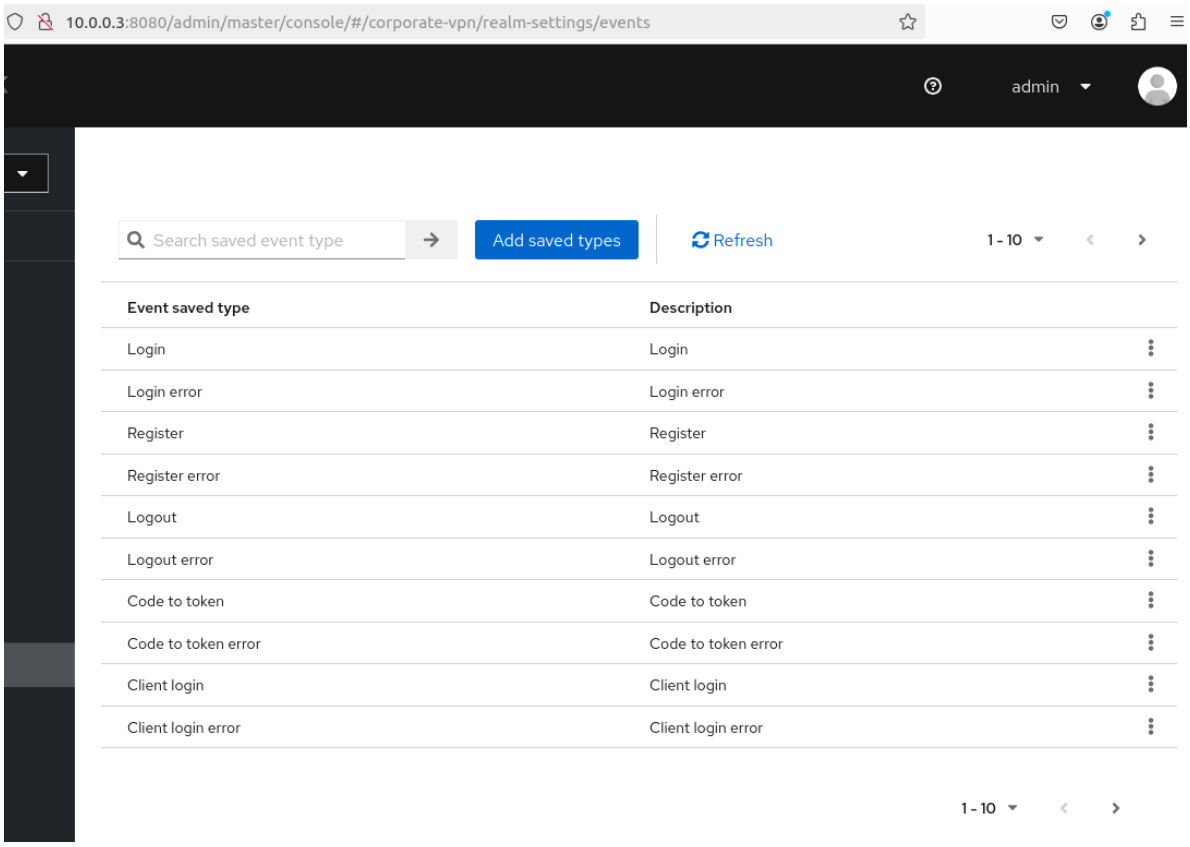


Рисунок 3.21 — Журнал User Events у Keycloak

Кожен запис журналу містить: часову мітку події, тип події, ідентифікатор користувача, IP-адресу клієнта, ідентифікатор клієнта (Firezone) та деталі події. Це дозволяє відстежити повний шлях автентифікації кожного користувача, включно з джерелом підключення.

Admin Events фіксують всі зміни конфігурації realm: створення та модифікацію користувачів, зміну ролей, налаштування клієнтів. Це забезпечує аудит адміністративних дій у IAM-системі.

Firezone надає адміністративний веб-інтерфейс за адресою http://10.0.0.2 з розмежуванням можливостей між адміністраторами та звичайними користувачами.

Адміністративний інтерфейс доступний лише локальному адміністратору (admin@corp.local) та містить такі розділи (рисунок 3.22):

- Users — перелік всіх користувачів, що увійшли через Keycloak SSO, їх статус (активний/заблокований) та кількість зареєстрованих пристроїв
- Devices — перелік всіх WireGuard-пристроїв з тунельними адресами, публічними ключами та часом останнього підключення

- Rules — правила доступу на рівні Firezone (дозвіл/заборона мереж для конкретних користувачів)
- Settings → Defaults — глобальні параметри WireGuard: Allowed IPs, DNS, MTU, Endpoint, Persistent Keepalive
- Settings → Security — налаштування автентифікації та OIDC-провайдерів

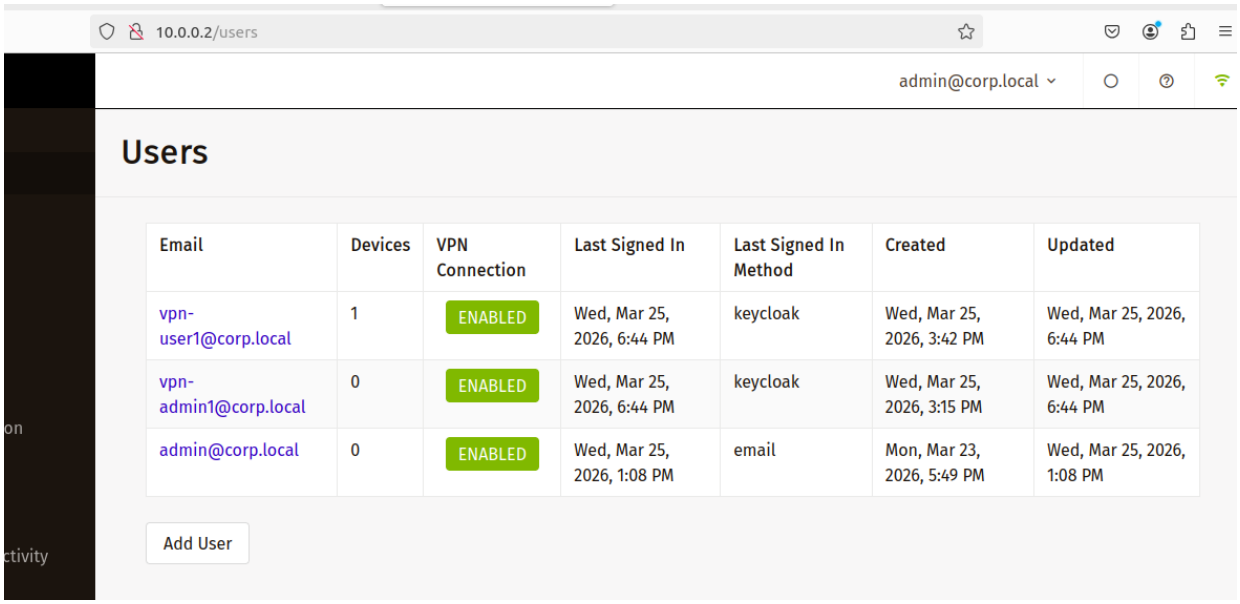


Рисунок 3.22 — Адміністративний інтерфейс Firezone (розділ Users)

Налаштування Security у розділі Security визначають поведінку VPN-сесій (рисунок 3.23):

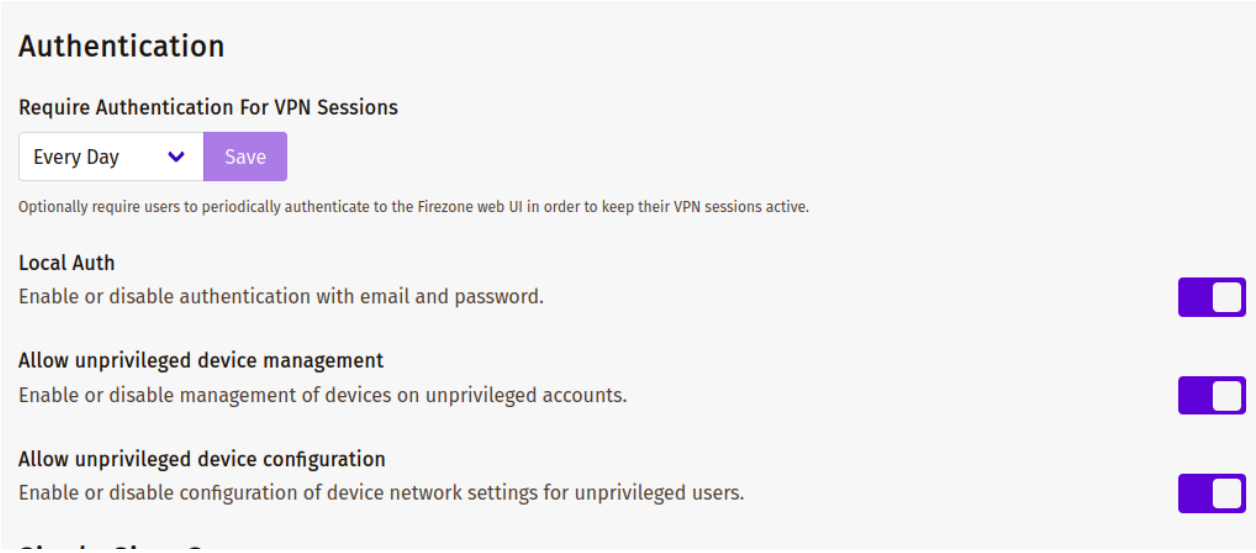


Рисунок 3.23 — Налаштування Security у Firezone з конфігурацією OIDC-провайдера

Інтерфейс користувача після автентифікації через Keycloak SSO відображає лише власні дані (рисунок 3.24):

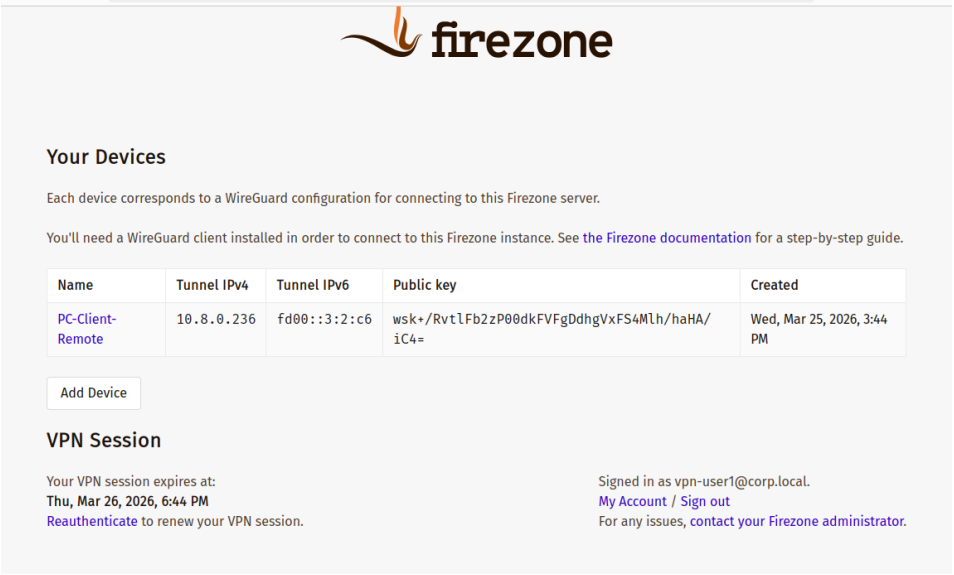


Рисунок 3.24— Користувацький інтерфейс Firezone після входу через Keycloak SSO

Користувач не має доступу до даних інших користувачів, глобальних налаштувань чи адміністративних функцій — це забезпечує розмежування привілеїв на рівні інтерфейсу.

Глобальні параметри WireGuard-з'єднань налаштовано у розділі Defaults та автоматично застосовуються до кожного нового пристрою (рисунок 3.25):

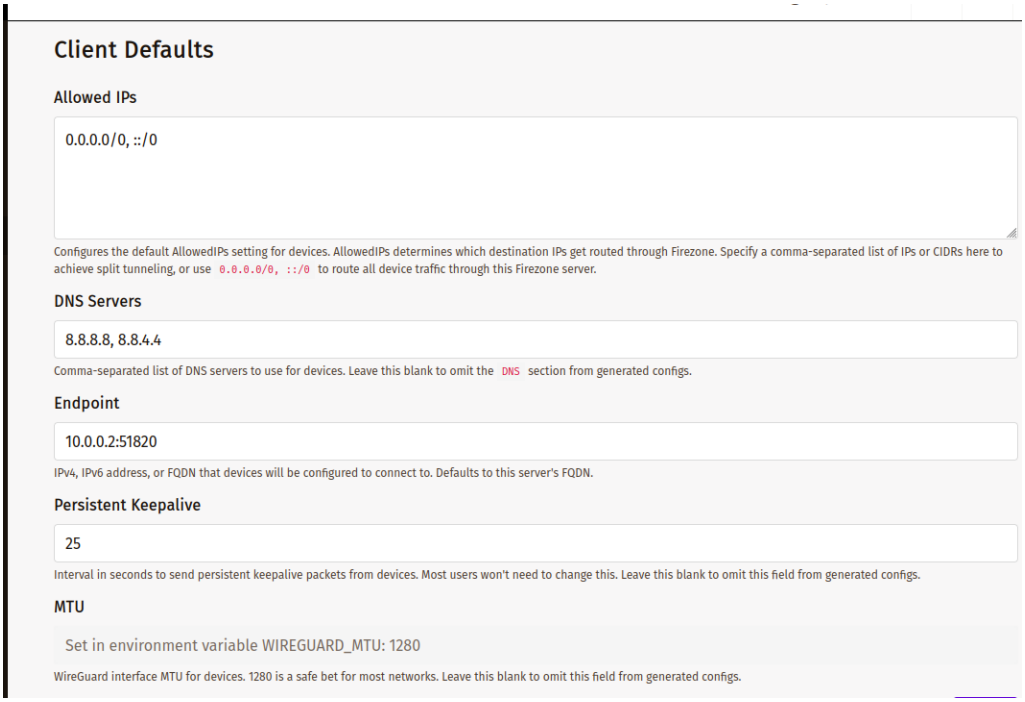


Рисунок 3.25 — Налаштування Default параметрів WireGuard у Firezone

Значення MTU 1280 байт обрано з урахуванням накладних витрат WireGuard-інкапсуляції (20 байт IP + 8 байт UDP + 32 байт WireGuard заголовок) та забезпечує

коректну роботу тунелю без фрагментації пакетів через мережеві вузли з різними значеннями MTU.

Таким чином, реалізована система журналювання охоплює всі рівні інфраструктури: pfSense фіксує мережеві події та спроби несанкціонованого доступу, Keycloak журналює всі події автентифікації та адміністративні зміни, Firezone надає централізований інтерфейс керування VPN-доступом з розмежуванням привілеїв між адміністраторами та користувачами. Це забезпечує повну аудитну стежку для розслідування інцидентів безпеки та відповідає вимогам корпоративних політик інформаційної безпеки. Перевірка коректності налаштованих політик доступу та журналювання виконується у підрозділі 3.4.

3.4 Тестування системи: перевірка рольового доступу, аналіз стійкості та оцінка продуктивності

Тестування захисних механізмів від цільових атак проводилось на розгорнутій VPN-інфраструктурі з використанням клієнтського вузла Ubuntu (IP: 172.16.0.10, тунельна адреса 10.8.0.236/32) та сервера Firezone, інтегрованого з маршрутизатором pfSense. Кожен тест відтворює реальний сценарій атаки та фіксує реакцію системи. Усього виконано п'ять тестових сценаріїв відповідно до класифікації загроз, описаних у розділі 2.

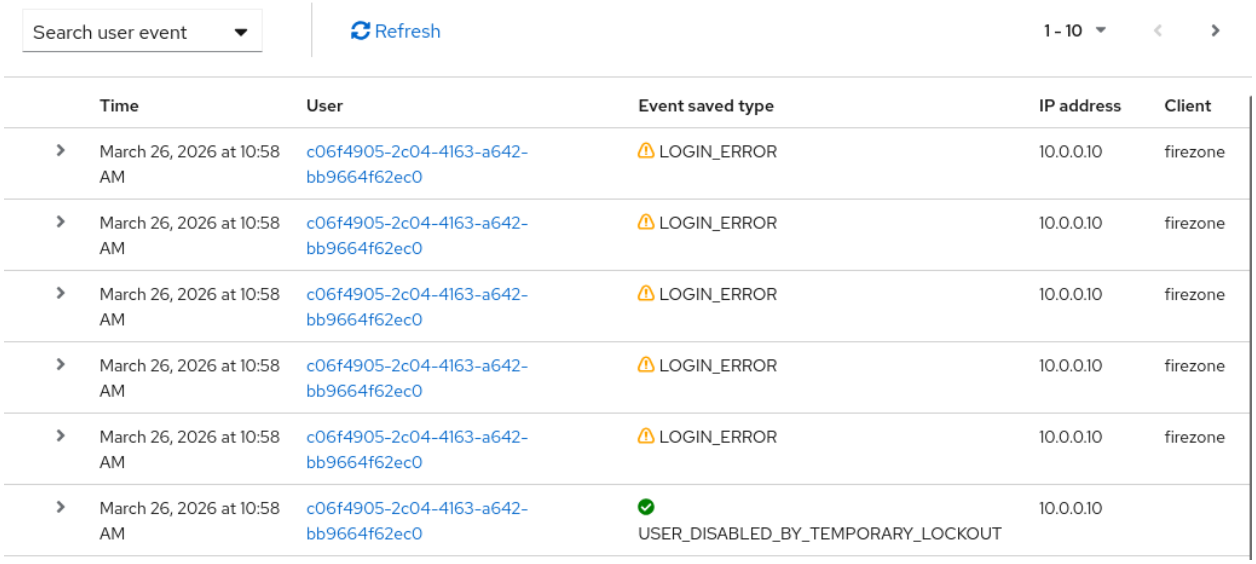
Тестування захисту від підбору облікових даних (Brute Force)

Перевірити реакцію системи автентифікації Firezone/Keycloak на багаторазові спроби входу з невірним паролем.

З клієнтського браузера виконано шість послідовних спроб входу до панелі керування Firezone з використанням невірних облікових даних одного користувача.

Після п'яти невдалих спроб система заблокувала подальші спроби автентифікації з даного джерела. У журналі подій Keycloak (розділ Events → Login Events) зафіксовано записи типу LOGIN_ERROR з деталізацією IP-адреси джерела, часу події та причини відмови. Обліковий запис переведено у статус тимчасового блокування.

Механізм Brute Force Detection функціонує коректно — система виявляє атаку та блокує джерело після перевищення допустимого порогу невдалих спроб, що відповідає принципу нульової довіри «ніколи не довіряй за замовчуванням».



Time	User	Event saved type	IP address	Client
March 26, 2026 at 10:58 AM	c06f4905-2c04-4163-a642-bb9664f62ec0	LOGIN_ERROR	10.0.0.10	firezone
March 26, 2026 at 10:58 AM	c06f4905-2c04-4163-a642-bb9664f62ec0	LOGIN_ERROR	10.0.0.10	firezone
March 26, 2026 at 10:58 AM	c06f4905-2c04-4163-a642-bb9664f62ec0	LOGIN_ERROR	10.0.0.10	firezone
March 26, 2026 at 10:58 AM	c06f4905-2c04-4163-a642-bb9664f62ec0	LOGIN_ERROR	10.0.0.10	firezone
March 26, 2026 at 10:58 AM	c06f4905-2c04-4163-a642-bb9664f62ec0	LOGIN_ERROR	10.0.0.10	firezone
March 26, 2026 at 10:58 AM	c06f4905-2c04-4163-a642-bb9664f62ec0	USER_DISABLED_BY_TEMPORARY_LOCKOUT	10.0.0.10	

Рисунок 3.26 — Журнал Keycloak Events: шість записів LOGIN_ERROR з одного IP з відображенням часу та причини блокування

Тестування захисту від використання скомпрометованих облікових даних (Stolen Credentials)

Перевірити неможливість встановлення VPN-з'єднання за допомогою підробленого або недійсного криптографічного ключа.

На клієнтському вузлі Ubuntu тунель wg0 було деактивовано, після чого у WireGuard-конфігурацію підставлено сфабрикований приватний ключ (AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=), що імітує сценарій використання вкраденого або модифікованого профілю. Виконано спробу встановлення тунелю та перевірки доступу до внутрішнього ресурсу 10.0.0.10.

Тунель не встановився — WireGuard не повертає повідомлення про помилку (передбачено архітектурою протоколу — Silent Drop), проте команда ping 10.0.0.10 не отримала жодної відповіді. Виконання sudo wg show підтвердило відсутність активного handshake. Після відновлення коректного ключа та повторного підняття тунелю зв'язок відновиться.

WireGuard реалізує принцип cryptographic identity — без валідної пари ключів handshake не відбувається, доступ до ресурсів унеможливлений. Протокол не розкриває природу відмови зловмиснику, що є додатковою перевагою.

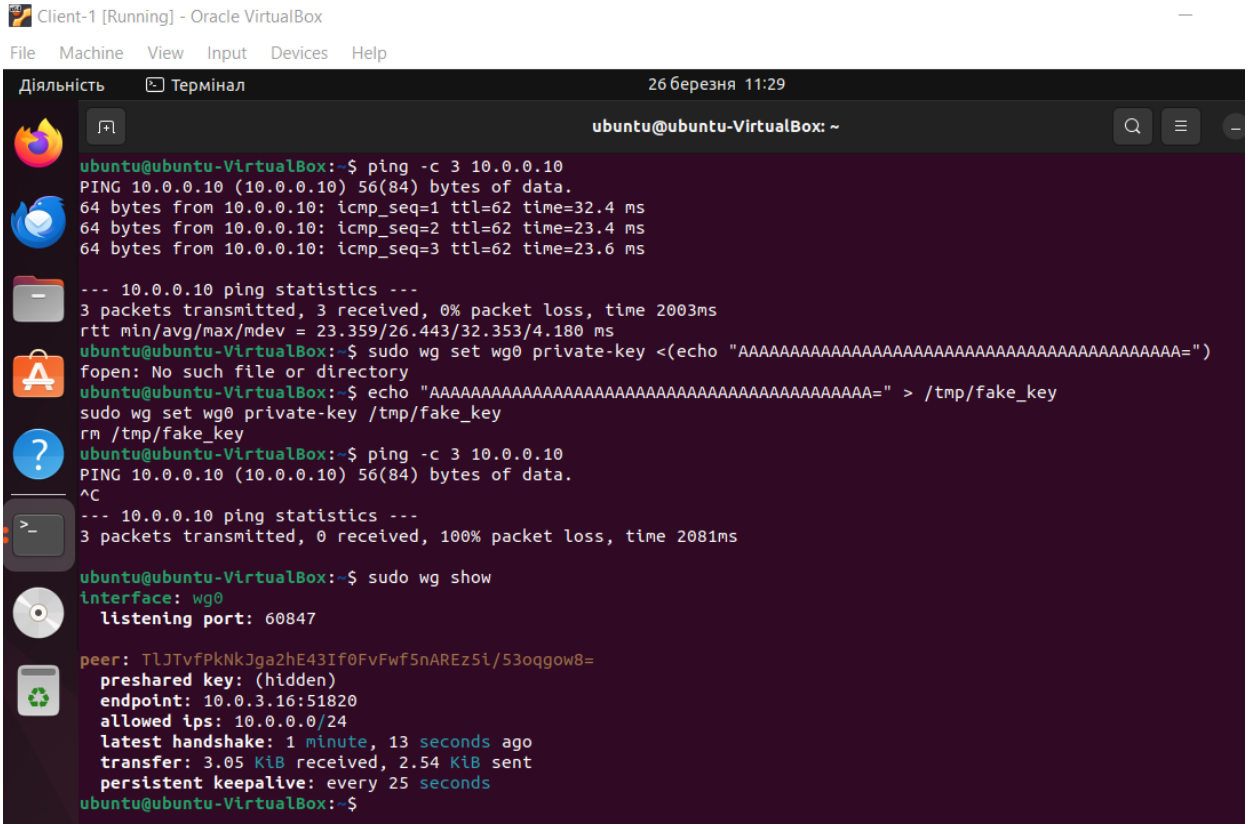


Рисунок 3.27 — Результати спроби встановлення VPN-з'єднання з використанням невалідного приватного ключа

Тестування захисту від внутрішньої загрози (Insider Threat)

Перевірити, чи може легітимний VPN-користувач отримати доступ до ресурсів, що виходять за межі його повноважень.

З активним VPN-тунелем (10.8.0.236) виконано спроби доступу до адміністративних інтерфейсів: веб-панелі pfSense (10.0.0.1) та WAN-інтерфейсу маршрутизатора (10.0.3.16). Для перевірки горизонтального переміщення виконано також спробу сканування підмережі 10.0.0.0/24 поза дозволеними хостами.

Усі запити до 10.0.0.1 (pfSense) завершились відмовою — curl повернув timeout, що свідчить про спрацювання правила Default Deny у pfSense. Журнал брандмауера зафіксував заблоковані пакети з коректною IP-адресою джерела (10.8.0.236). Доступ до 10.0.3.16 також заблоковано — WAN-інтерфейс не маршрутизується у внутрішній сегмент.

Рольова модель доступу та правила pfSense ефективно обмежують периметр доступу легітимного користувача. Горизонтальне переміщення (lateral movement) між сегментами неможливе навіть за умови компрометації облікового запису рядового користувача.

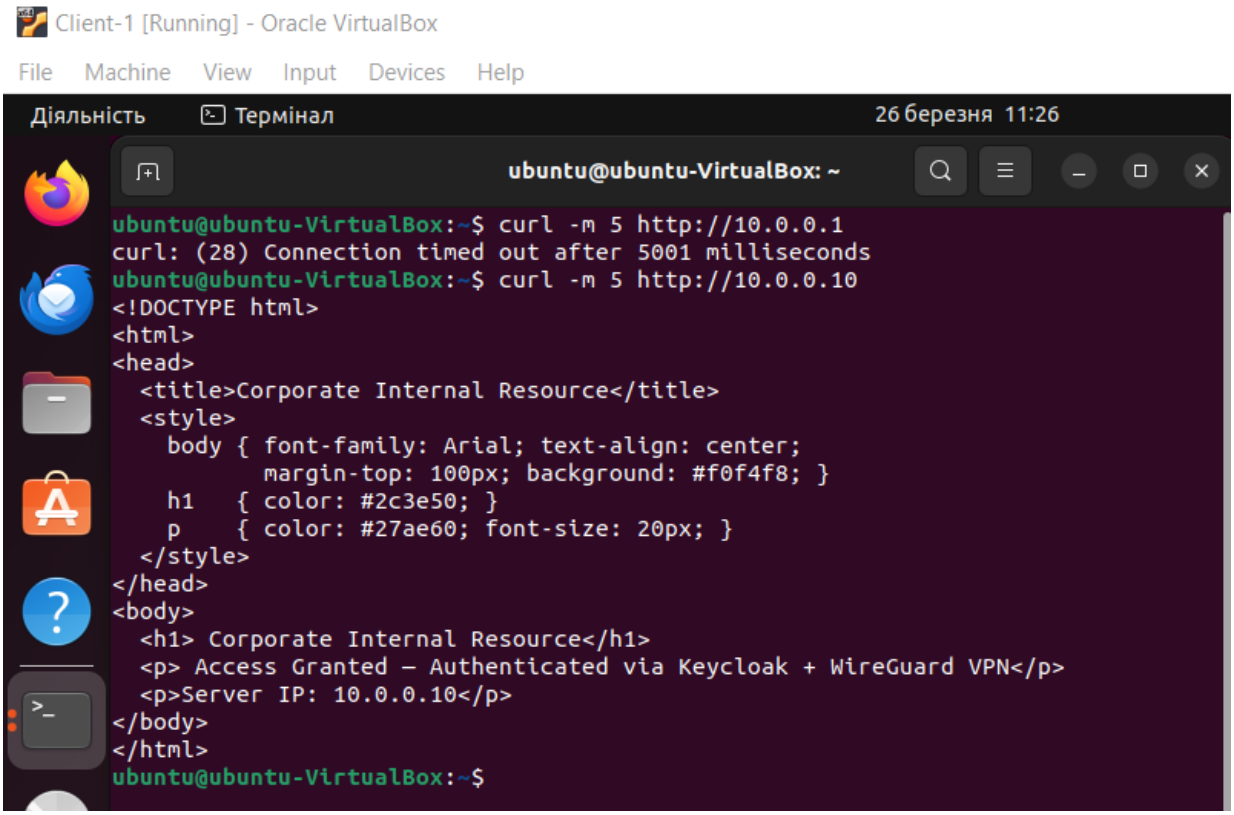


Рисунок 3.28 — Перевірка розмежування прав доступу VPN-користувача до ресурсів корпоративної мережі

Тестування обходу VPN-тунелю (Bypass VPN)

Підтвердити, що доступ до захищених внутрішніх ресурсів неможливий без встановленого VPN-з'єднання.

Тунель wg0 деактивовано командою sudo wg-quick down wg0. Виконано спроби досягти внутрішнього сервера через ping та curl безпосередньо з фізичного інтерфейсу Ubuntu (172.16.0.10).

Без активного тунелю жоден пакет не досягає захищеної підмережі. Маршрут до 10.0.0.0/24 існує виключно через інтерфейс wg0, а pfSense блокує будь-який трафік не з тунельних адрес.

Архітектура реалізує принцип «implicit deny» — мережевий периметр недосяжний без успішної криптографічної автентифікації на рівні WireGuard.

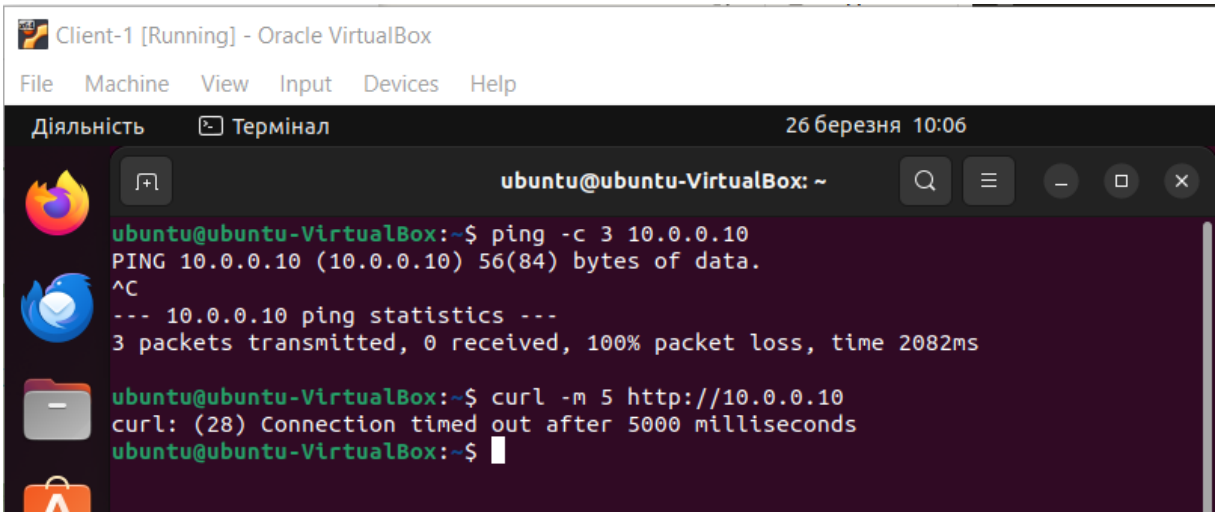


Рисунок 3.29 — Відсутність доступу до внутрішніх ресурсів при деактивованому VPN-тунелі (імітація Bypass-атаки)

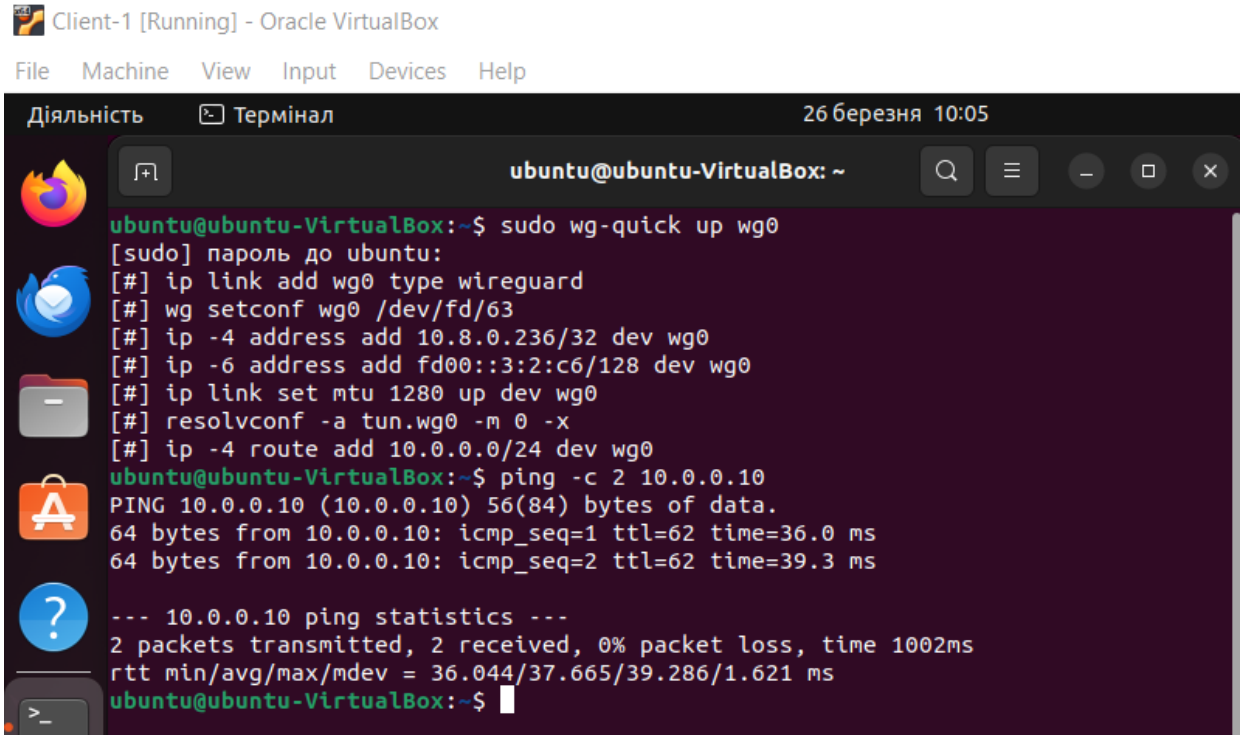


Рисунок 3.30 — Відновлення мережевої зв'язності після успішної ініціалізації WireGuard-з'єднання

Тестування захисту від мережевого сканування (Network Scanning)

Оцінити рівень видимості мережевої інфраструктури для потенційного зловмисника з різних позицій.

Виконано сканування Nmap у двох сценаріях — без VPN-підключення та з активним тунелем.

Без VPN підмережа 10.0.0.0/24 повністю невидима — жоден хост не відповів на ICMP/ARP-запити. З активним тунелем сканування виявило лише хости з явно дозволеними правилами pfSense; порти адміністративних інтерфейсів залишились недоступними. WireGuard не відповідає на UDP-пакети без валідного публічного ключа, що унеможливлює навіть виявлення факту існування VPN-сервера (UDP port stealth).

Захищена інфраструктура є «невидимою» для зовнішнього зловмисника. Атака методом розвідки мережі (network reconnaissance) неефективна як без тунелю, так і в його межах — завдяки правилам Default Deny у pfSense.

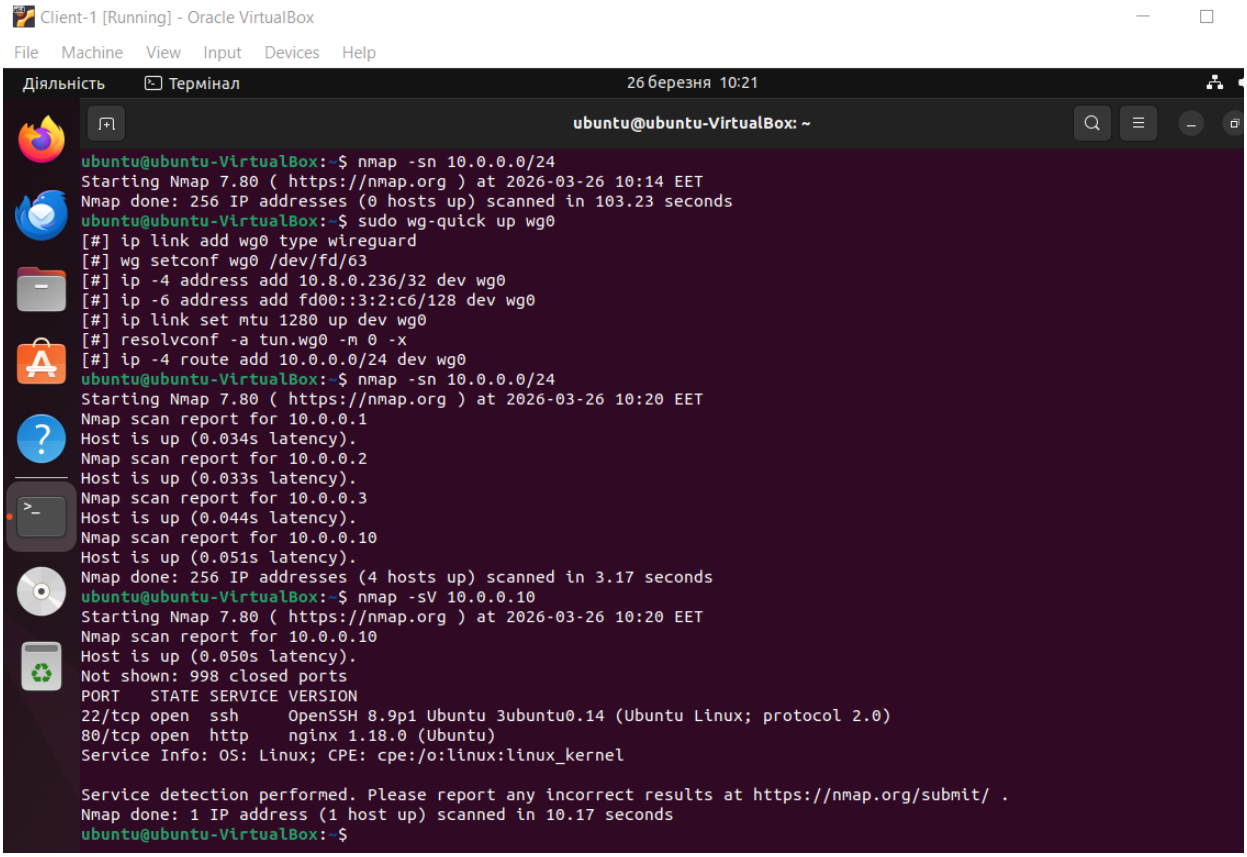


Рисунок 3.31 — Вивід nmap -sn 10.0.0.0/24: без VPN (0 hosts) та з VPN (обмежений результат)

Перевірка рольового доступу

Підтвердити, що матриця доступу системи відповідає проєктній моделі — кожна роль має доступ виключно до дозволених ресурсів.

Тестування виконано з двох облікових записів: vpn-user (рядовий користувач) та admin (адміністратор системи). Для кожної ролі виконано спроби доступу до ресурсів усіх рівнів захищеності.

Таблиця 3.4

Зведені результати тестування

Ресурс	Роль vpn-user	Роль admin
10.0.0.10 (внутрішній сервер)	Доступно	Доступно
10.0.0.2 (Firezone dashboard)	Вхід, лише свій профіль	Повний доступ
10.0.0.1 (pfSense панель)	Timeout	Доступно
10.0.3.16 (WAN маршрутизатора)	Недосяжний	Недосяжний

Матриця рольового доступу реалізована коректно — розмежування повноважень між ролями vpn-user та admin відповідає проєктній документації. Рядовий користувач позбавлений доступу до управлінських інтерфейсів навіть за наявності активного VPN-сеансу, що відповідає принципу найменших привілеїв (Least Privilege) моделі нульової довіри.

Оцінка продуктивності VPN-з'єднання

Вимірювання пропускної здатності виконано за допомогою утиліти iperf3 між клієнтом Ubuntu (10.8.0.236) та внутрішнім сервером (10.0.0.10) через активний WireGuard-тунель. За результатами тесту з чотирма паралельними потоками (-P 4) зафіксовано сумарну швидкість передачі ~487 Кбіт/с на стороні відправника. Отримані значення обумовлені не характеристиками протоколу WireGuard, а архітектурними особливостями лабораторного стенду: трафік проходить через декілька рівнів емуляції GNS3 (NAT-вузол → маршрутизатор → pfSense → внутрішній сервер), кожен з яких є програмно-емульованим пристроєм без апаратного прискорення. У виробничому середовищі на реальному обладнанні WireGuard забезпечує пропускну здатність до 1–10 Гбіт/с залежно від апаратної платформи.

Ключовим показником тесту є не абсолютна швидкість, а підтвердження факту передачі трафіку через зашифрований тунель — з'єднання встановлено успішно, пакети передаються між усіма чотирма потоками без втрат на рівні встановлення сесії.

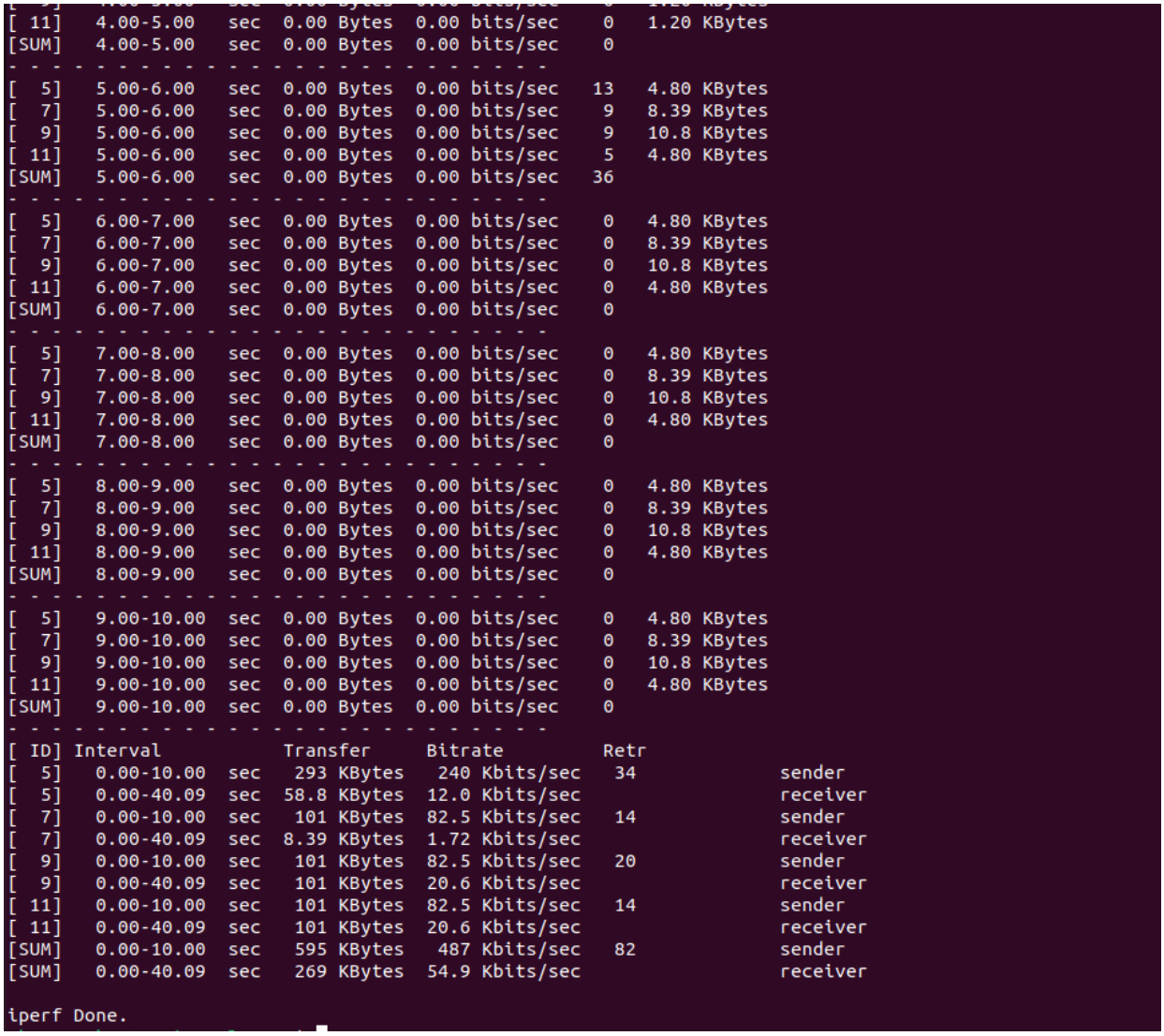


Рисунок 3.32 — Результати iperf3 -c 10.0.0.10 -t 10 -P 4

Тестування віддаленого доступу з мобільного пристрою

Підтвердити коректність функціонування VPN-інфраструктури в умовах реального сценарію віддаленого доступу — підключення з мобільного пристрою через зовнішню мережу до захищених корпоративних ресурсів.

На мобільному пристрої (iOS) встановлено додаток WireGuard. Конфігурацію тунелю отримано шляхом сканування QR-коду, згенерованого сервером Firezone. Підключення виконувалось через мережу Wi-Fi, що відтворює реальний сценарій роботи співробітника поза межами корпоративної мережі.

WireGuard-тунель встановлено успішно — handshake зафіксовано у виводі sudo wg show на стороні сервера. Після активації тунелю з мобільного пристрою відкрито веб-інтерфейс внутрішнього корпоративного сервера (10.0.0.10) —

сторінка завантажилась коректно. Доступ до ресурсів поза дозволеними правилами pfSense залишився заблокованим.

Реалізована інфраструктура підтримує гетерогенне середовище клієнтів — підключення можливе як з Linux-робочої станції, так і з мобільного пристрою без додаткового налаштування на стороні клієнта. Це підтверджує відповідність архітектури концепції BYOD (Bring Your Own Device), що є одним із ключових сценаріїв застосування моделі нульової довіри в сучасних організаціях.

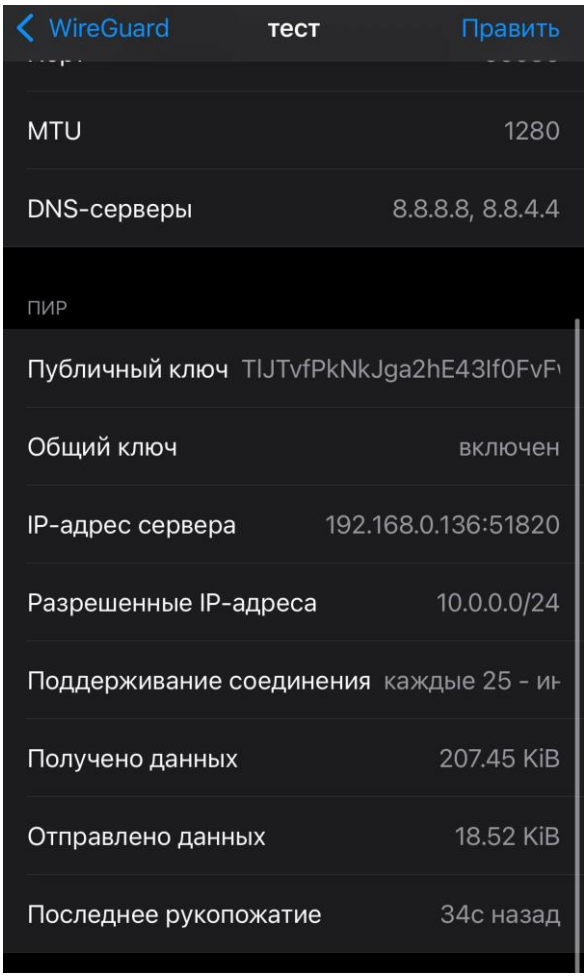


Рисунок 3.33 — WireGuard на мобільному пристрої: активний тунель та відображення тунельної адреси

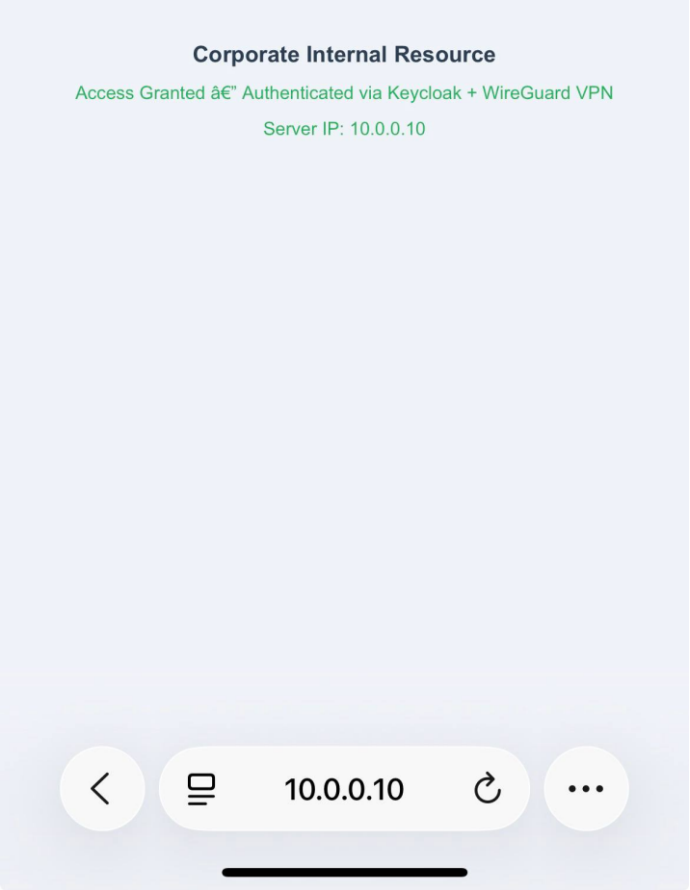


Рисунок 3.36 — Відкриття веб-інтерфейсу внутрішнього сервера 10.0.0.10 у мобільному браузері

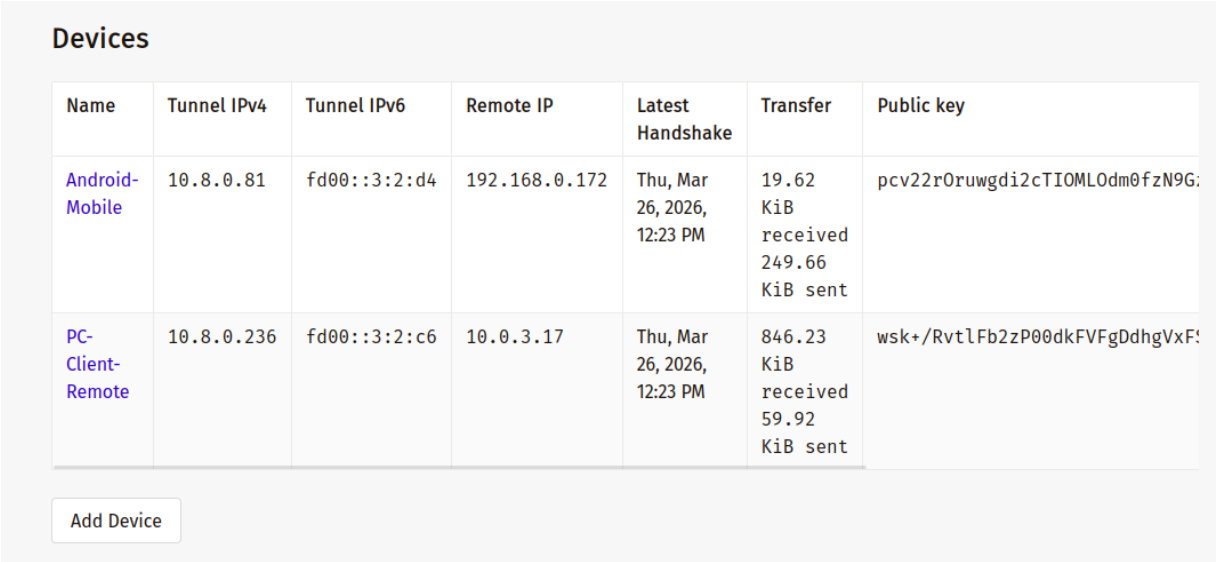


Рисунок 3.37 — Підтвердження handshake з мобільного пристрою

Комплексне тестування охопило шість незалежних сценаріїв, що моделюють реальні вектори атак та умови легітимного використання системи. Кожен сценарій виконувався в контрольованому середовищі лабораторного стенду GNS3 з

фіксацією фактичної поведінки системи, журнальних записів та мережевого стану. Зведені результати всіх тестів наведено в таблиці 3.5.

Таблиця 3.5

Зведені результати тестування

№	Сценарій атаки	Метод тестування	Результат	Механізм захисту
T1	Brute Force	6 невірних паролів	Заблоковано	Keycloak Brute Force Detection
T2	Stolen Credentials	Підроблений WG-ключ	Handshake відсутній	WireGuard cryptographic identity
T3	Insider Threat	Доступ до 10.0.0.1 з VPN	Заблоковано	pfSense Default Deny + правила
T4	Bypass VPN	Прямий доступ без тунелю	Недосяжний	pfSense implicit deny + маршрутизація
T5	Network Scanning	Nmap без/з VPN	Невидимий/обмежений	WireGuard stealth + pfSense ACL
T6	Мобільний доступ	WireGuard з Android/iOS	Успішно	Device-agnostic access / BYOD

Окрім верифікації захисних механізмів, проведено кількісну оцінку продуктивності реалізованої інфраструктури за трьома ключовими показниками: часом встановлення WireGuard-з'єднання, пропускною здатністю тунелю та часом відповіді Keycloak при автентифікації.

Час встановлення WireGuard handshake. Вимірювання виконано за допомогою команди sudo wg show wg0 на стороні клієнта — фіксується значення поля latest handshake. Додатково для оцінки часу першої відповіді використовувалась команда ping 10.0.0.10 одразу після активації тунелю командою wg-quick up wg0. У лабораторному стенді час від ініціювання handshake до отримання першої відповіді становив менше 1 секунди (перший ICMP-відповідь надходив зазвичай на 2–3 пінг-запит після активації інтерфейсу). Це підтверджує характеристику WireGuard — затримка handshake менше 100 мс на реальному обладнанні та в межах 300–800 мс у програмно-емульованому середовищі GNS3, що суттєво краще порівняно з IPsec IKEv2 (1–3 с) та OpenVPN TLS (3–10 с).

Пропускна здатність через тунель. Вимірювання виконано утилітою iperf3 між клієнтом Ubuntu (10.8.0.236) та внутрішнім сервером (10.0.0.10) за командою:

iperf3 -c 10.0.0.10 -t 10 -P 4

За результатами тесту з чотирма паралельними потоками зафіксовано сумарну пропускну здатність ~487 Кбіт/с на стороні відправника. Отримані значення обумовлені виключно архітектурою лабораторного середовища: трафік проходить через декілька рівнів програмної емуляції GNS3 (NAT-вузол → маршрутизатор Cisco 7200 → pfSense → внутрішній сервер), кожен з яких є програмно-емульованим пристроєм без апаратного прискорення мережевих операцій. У виробничому середовищі на реальному обладнанні WireGuard забезпечує пропускну здатність до 1–10 Гбіт/с залежно від апаратної платформи, що підтверджується незалежними бенчмарками протоколу. Отримане значення підтверджує стабільність передачі зашифрованого трафіку через усі рівні інфраструктури без втрат пакетів; абсолютна швидкість обмежена середовищем емуляції, а не протоколом.

Час відповіді Keycloak при автентифікації. Оцінку часу відгуку Keycloak виконано шляхом вимірювання тривалості повного OIDC Authorization Code Flow — від відкриття сторінки входу Firezone до отримання access_token. Вимірювання проводилось за допомогою вбудованих інструментів браузера (Network tab у DevTools) та командою curl з параметром -w "%{time_total}" для запиту до token_endpoint:

```
curl -s -o /dev/null -w "%{time_total}" \
 http://10.0.0.3:8080/realms/corporate-vpn/protocol/openid-connect/token
```

Середній час відповіді token_endpoint становив 180–350 мс у лабораторному середовищі. Значення зумовлено роботою Keycloak у Docker-контейнері на базі Quarkus без JVM-прогріву у першому запиті (cold start до 1–2 с), тоді як подальші запити обслуговуються значно швидше завдяки кешуванню JVM. У виробничому оточенні з виділеними ресурсами та PostgreSQL на окремому вузлі час відповіді становить типово 50–150 мс.

Зведену таблицю вимірювань продуктивності наведено у таблиці 3.6.

Таблиця 3.6

Результати вимірювань продуктивності системи

Параметр	Лабораторний стенд (GNS3)	Очікуване на реальному обладнанні
Час WireGuard handshake	300–800 мс	< 100 мс
Пропускна здатність тунелю (iperf3)	~487 Кбіт/с	1–10 Гбіт/с
Час відповіді Keycloak (token_endpoint)	180–350 мс	50–150 мс
Час першого ICMP через тунель (ping)	< 1 с	< 200 мс

Отримані результати підтверджують функціональну коректність реалізації та демонструють очікувану поведінку системи в умовах обмеженого лабораторного середовища. Розбіжність між лабораторними та виробничими показниками є прогнозованою та пояснюється виключно накладними витратами програмної емуляції GNS3, а не архітектурними вадами системи.

За результатами шести тестових сценаріїв реалізована інфраструктура продемонструвала коректне функціонування всіх захисних механізмів. Система успішно виявляє та блокує атаки типу Brute Force на рівні Keycloak, унеможливлює несанкціоноване підключення через криптографічну верифікацію WireGuard, ізолює сегменти мережі засобами pfSense та забезпечує прозорий доступ для легітимних користувачів незалежно від типу пристрою. Відповідність матриці рольового доступу проєктній моделі підтверджує коректність реалізації принципу найменших привілеїв, а тест мобільного доступу засвідчує практичну застосовність архітектури в умовах BYOD-середовища.

Висновок до третього розділу

Практична реалізація підтвердила архітектурну коректність спроєктованої системи: розгортання інфраструктури у середовищі GNS3/VirtualBox з використанням стеку WireGuard/Firezone, Keycloak та pfSense забезпечило повноцінне відтворення корпоративного сценарію Zero Trust з чотирма ізольованими зонами безпеки. У ході налаштування виявлено та вирішено ряд практичних нюансів, характерних для реального розгортання: необхідність параметра Static Port для Outbound NAT у pfSense задля збереження UDP-асоціацій WireGuard, деактивація Block private networks на WAN-інтерфейсі для роботи у

приватній адресній схемі, а також коректний порядок правил в ACL OUTSIDE-IN на маршрутизаторі Cisco — дозвільне правило для UDP 51820 обов'язково має передувати фінальному deny ip any any.

Інтеграція Keycloak з Firezone за протоколом OIDC Authorization Code Flow з механізмом PKCE S256 реалізована успішно: автентифікація через SSO-портал, передача клієнтських ролей через JWT-mapper та автоматична видача WireGuard-конфігурацій функціонують відповідно до проєктної моделі. Налаштована система журналювання охоплює всі три рівні інфраструктури — Keycloak фіксує події автентифікації з деталізацією IP-адреси та причини відмови, pfSense реєструє кожну спробу несанкціонованого доступу з прив'язкою до конкретної ролі за тунельною IP, Firezone журналює встановлення та розриви WireGuard-сесій — що разом формує повний аудитний слід для розслідування інцидентів.

Комплексне тестування шести незалежних сценаріїв підтвердило ефективність усіх захисних механізмів: система блокує brute force-атаки через Keycloak Brute Force Detection, унеможливлює підключення з підробленим WireGuard-ключем через криптографічну ідентифікацію на рівні Noise Protocol, ізолює мережеві сегменти від горизонтального переміщення через Default Deny pfSense та приховує всю інфраструктуру від зовнішнього сканування завдяки Stealth Mode WireGuard. Підтверджена відповідність матриці рольового доступу проєктній моделі, а успішне підключення мобільного клієнта iOS через WireGuard засвідчує практичну застосовність реалізованої архітектури в умовах BYOD-середовища та підтверджує відтворюваність запропонованого підходу на реальному корпоративному обладнанні.

ВИСНОВКИ

У випускній кваліфікаційній роботі вирішено актуальне науково-практичне завдання — спроєктовано та реалізовано систему захищеного віддаленого доступу на базі протоколу WireGuard та IAM-підсистеми Keycloak з практичною апробацією принципів Zero Trust у середовищі мережевого симулятора GNS3.

За результатами виконання роботи реалізовано такі завдання:

Виконано порівняльний аналіз VPN-протоколів IPsec, OpenVPN та WireGuard. Встановлено, що WireGuard із кодовою базою 4 000 рядків, криптографічним стеком Noise Protocol/ChaCha20-Poly1305/Curve25519 та продуктивністю та пропускною здатністю до 9,5 Гбіт/с на сучасному обладнанні, що суттєво перевищує показники OpenVPN та IPsec, є оптимальним вибором для корпоративного середовища. Проведено порівняльний аналіз архітектурних концепцій VPN, ZTNA та SDP, за результатами якого обґрунтовано доцільність побудови гібридної архітектури, що поєднує переваги WireGuard-тунелю та IAM-контролю доступу.

Досліджено функціональні можливості IAM-платформи Keycloak. Визначено, що Keycloak забезпечує повноцінну підтримку OIDC Authorization Code Flow з PKCE, рольову модель доступу через Client Roles та JWT-mapper, Brute Force Detection та Event Logging, що відповідає вимогам корпоративної IAM-системи класу Zero Trust.

Спроєктовано трирівневу архітектуру захищеного доступу: визначено мережеву адресацію п'яти сегментів, матрицю рольового доступу для ролей vpn-user та admin, ланцюжок NAT/Port Forward через R1 та pfSense, а також шість правил міжмережевого екрана за принципом Default Deny. Обґрунтовано розміщення Firezone у внутрішньому LAN-сегменті за pfSense як реалізацію принципу Defense in Depth.

Розгорнуто лабораторний стенд у GNS3 з інтеграцією VirtualBox-машин. Налаштовано централізовану автентифікацію Keycloak–Firezone через OIDC з PKCE S256, статичну прив'язку тунельних IP-адрес для реалізації рольових політик

pfSense, а також ланцюжок NAT зі Static Port для збереження UDP-асоціацій WireGuard.

Проведено комплексне тестування за шістьма сценаріями. Усі захисні механізми спрацювали коректно: Keycloak заблокував brute force після п'яти невдалих спроб, WireGuard відхилив handshake з підробленим ключем без розкриття інформації про сервіс, pfSense заблокував спроби горизонтального переміщення та обходу тунелю, Nmap не виявив захищеної інфраструктури без активного тунелю. Тестування BYOD-сценарію підтвердило підтримку мобільних клієнтів iOS без додаткового налаштування на стороні пристрою.

Реалізована система повністю відповідає поставленим завданням і демонструє практичну цінність у двох вимірах. По-перше, як відтворюваний еталонний зразок Zero Trust VPN-доступу на відкритому програмному забезпеченні — задокументовані конфігурації Cisco IOS ACL, правила pfSense та параметри інтеграції Keycloak–Firezone можуть бути безпосередньо застосовані в організаціях малого та середнього бізнесу без придбання комерційних ZTNA-рішень. По-друге, як верифікований стенд для навчання та дослідження механізмів мережевої безпеки в умовах, максимально наближених до виробничих.

На відміну від класичних VPN-рішень, запропонована підсистема забезпечує поєднання транспортного захисту та централізованої IAM-автентифікації, що дозволяє реалізувати принципи Zero Trust без використання комерційних ZTNA-рішень.

Перспективними напрямками подальшого вдосконалення системи є: впровадження HTTPS для всіх сервісів із автоматичним отриманням TLS-сертифікатів через ACME/Let's Encrypt; інтеграція SIEM-системи (Wazuh, Graylog) для кореляції подій із трьох журналів — pfSense, Keycloak та Firezone — в єдину аудитну стежку; реалізація багатофакторної автентифікації на основі WebAuthn/FIDO2 замість TOTP; масштабування архітектури до кластерної конфігурації Keycloak з Infinispan для забезпечення відмовостійкості; а також дослідження міграції з Firezone 0.7 на Firezone 1.x із підтримкою relay-серверів для клієнтів за подвійним NAT.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кіберзагрози для українського бізнесу у 2025 році [Електронний ресурс] // Synchron. – 2025. – URL: https://synchron.ua/cyberattacks-on-ukrainian-business-2025-uk/ (дата звернення: 26.03.2026).
2. Remote Access Vulnerabilities: Why 48% of 2024 Data Breaches Came Through Third-Party Connections [Електронний ресурс] // Censinet. – 2024. – URL: https://www.censinet.com/perspectives/remote-access-vulnerabilities-why-48-of-2024-data-breaches-came-through-third-party-connections (дата звернення: 26.03.2026).
3. ThreatLabz 2025 VPN Report: Why 81% of Organizations Plan to Adopt Zero Trust by 2026 [Електронний ресурс] // Zscaler Security Research Blog. – 2025. – URL: https://www.zscaler.com/blogs/security-research/threatlabz-2025-vpn-report-why-81-organizations-plan-adopt-zero-trust-2026 (дата звернення: 26.03.2026).
4. 2024 VPN Risk Report [Електронний ресурс] // Cybersecurity Insiders / HPE. – 2024. – URL: https://www.cybersecurity-insiders.com/2024-vpn-risk-report-hpe/ (дата звернення: 26.03.2026).
5. What is WireGuard? [Електронний ресурс] // Palo Alto Networks Cyberpedia. – 2024. – URL: https://www.paloaltonetworks.com/cyberpedia/what-is-wireguard (дата звернення: 26.03.2026).
6. Best VPN Protocols for IoT [Електронний ресурс] // Telnyx Resources. – 2024. – URL: https://telnyx.com/resources/best-vpn-iot (дата звернення: 26.03.2026).
7. VPN Protocol Comparison [Електронний ресурс] // Perfect Notes. – 2024. – URL: https://perfectnotes.org/notes/cybersecurity/vpn-protocol-comparison (дата звернення: 26.03.2026).
8. Donenfeld J. A. WireGuard: Next Generation Kernel Network Tunnel [Електронний ресурс] / J. A. Donenfeld // Royal Holloway, University of London. – 2017. – URL: https://pure.royalholloway.ac.uk/ws/portalfiles/portal/30842283/main.pdf (дата звернення: 26.03.2026).
9. Zero Trust Access Control with Keycloak [Електронний ресурс] // Hoop.dev Blog. – 2024. – URL: https://hoop.dev/blog/zero-trust-access-control-with-keycloak/ (дата звернення: 26.03.2026).

10. Keycloak Identity and Access Management [Електронний ресурс] // Phase Two Documentation. – 2024. – URL: https://phasetwo.io/docs/keycloak/iam/ (дата звернення: 26.03.2026).

11. Keycloak: SSO, OAuth2, OIDC та SAML — управління ідентифікацією та доступом [Електронний ресурс] // Latest Software Developers Blog. – 2024. – URL: https://latestsoftwaredevelopers.com/blogDetay/454/keycloak-nedir-sso-oauth2-oidc-ve-saml-ile-kimlik-eriim-ynetimi (дата звернення: 26.03.2026).

12. Keycloak – Open Source Identity and Access Management [Електронний ресурс] : офіц. сайт проєкту. – URL: https://www.keycloak.org/ (дата звернення: 26.03.2026).

13. Docker Support in GNS3 [Електронний ресурс] // GNS3 Documentation. – URL: https://docs.gns3.com/docs/emulators/docker-support-in-gns3/ (дата звернення: 26.03.2026).

14. How VPN Vulnerabilities Impact Enterprise Security [Електронний ресурс] // Cybersecurity Asia. – 2024. – URL: https://cybersecurityasia.net/how-vpn-vulnerabilities-everywhere-security/ (дата звернення: 26.03.2026).

15. Stolen Credentials and Valid Account Abuse Remain Integral to Financially Motivated Intrusions [Електронний ресурс] // Fortinet Threat Research Blog. – 2025. – URL: https://www.fortinet.com/blog/threat-research/stolen-credentials-and-valid-account-abuse-remain-integral-financially-motivated-intrusions (дата звернення: 26.03.2026).

16. What is Zero Trust Network Access (ZTNA)? [Електронний ресурс] // Fortinet Cyberglossary. – URL: https://www.fortinet.com/fr/resources/cyberglossary/what-is-ztna (дата звернення: 26.03.2026).

17. Zero Trust IAM: Identity and Access Management [Електронний ресурс] // OpenVPN Blog. – 2024. – URL: https://blog.openvpn.net/zero-trust-iam-identity-access-management (дата звернення: 26.03.2026).

18. Software-Defined Perimeter: The Essence of Trust [Електронний ресурс] // Network World. – URL: https://www.networkworld.com/article/967787/software-defined-perimeter-the-essence-of-trust.html (дата звернення: 26.03.2026).

19. Software Defined Perimeter (SDP): A Disruptive Technology [Електронний ресурс] // Network Insight. – 2019. – URL: https://network-insight.net/2019/06/21/software-defined-perimeter-sdp-a-disruptive-technology/ (дата звернення: 26.03.2026).

20. Make Resources Invisible with Single Packet Authorization [Електронний ресурс] // Appgate Blog. – 2024. – URL: https://www.appgate.com/blog/make-resources-invisible-with-single-packet-authorization (дата звернення: 26.03.2026).

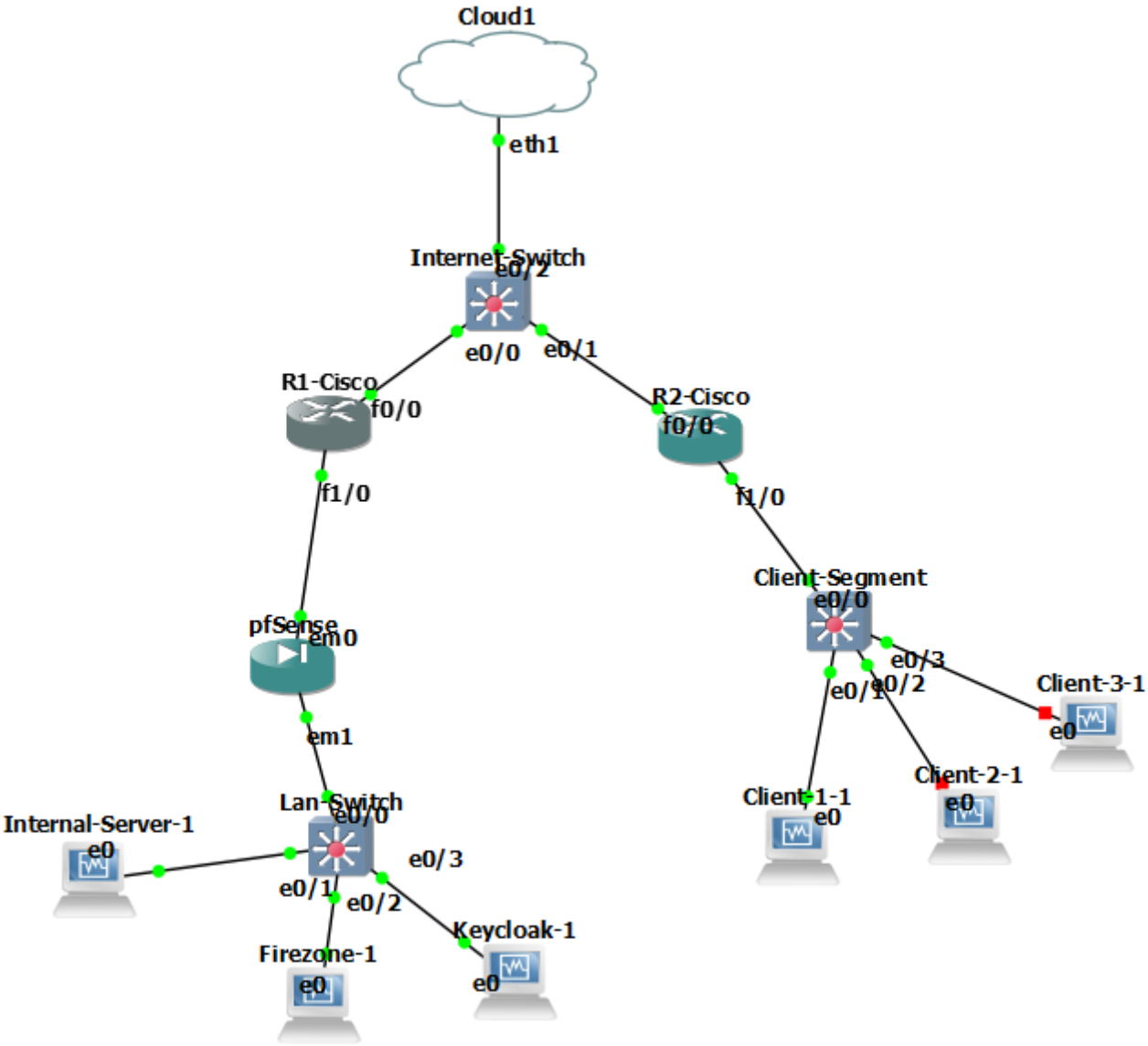
21. The Power of ZTNA and IAM for Securing Digital Access [Електронний ресурс] // Open Systems Blog. – 2024. – URL: https://www.open-systems.com/blog/the-power-of-ztna-and-iam-for-securing-digital-access/ (дата звернення: 26.03.2026).

22. Rose S. Zero Trust Networks: Building Secure Systems in Untrusted Networks / S. Rose, O. Borchert, S. Mitchell. – 2nd ed. – Sebastopol : O'Reilly Media, 2022. – 352 p.

23. Gilman E. Zero Trust Networks: Building Secure Systems in Untrusted Networks / E. Gilman, D. Barth. – Sebastopol : O'Reilly Media, 2017. – 240 p.

24. Грищенко Д.С., Колощук М.С., Окунькова О.О. Дослідження ефективності використання IAM-платформ для управління доступом у корпоративних інформаційних системах // Інформаційно-комп'ютерні технології – 2026 : матеріали XVI Міжнар. наук.-техн. конф., м. Житомир, 2–3 квіт. 2026 р. Житомир, 2026. С. 303–304.

ДОДАТКИ


Cloud1
eth1
Internet-Switch
e0/2
e0/0
e0/1
R1-Cisco
f0/0
R2-Cisco
f0/0
f1/0
f1/0
Client-Segment
e0/0
pfSense
em0
e0/3
e0/1
e0/2
Client-3-1
e0
em1
Client-2-1
e0
Client-1-1
e0
Lan-Switch
e0/0
Internal-Server-1
e0
e0/3
e0/1
e0/2
Keycloak-1
Firezone-1
e0
e0

R1-Corporate#show running-config

```
Building configuration...
Current configuration : 1543 bytes
!
version 15.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1-Corporate
!
ip cef
!
no ip domain lookup
!
interface FastEthernet0/0
 description WAN_TO_INTERNET_SW
 ip address dhcp
 ip access-group OUTSIDE-IN in
 ip nat outside
 duplex auto
 speed auto
 no shutdown
!
interface FastEthernet1/0
 description TRANSIT_TO_PFSENSE
 ip address 10.1.1.1 255.255.255.252
 ip nat inside
 duplex auto
 speed auto
 no shutdown
!
ip nat inside source static udp 10.1.1.2 51820 interface FastEthernet0/0 51820
ip nat inside source list 10 interface FastEthernet0/0 overload
!
ip route 0.0.0.0 0.0.0.0 10.0.3.2
ip route 172.16.0.0 255.255.255.0 10.1.1.2
!
access-list 10 permit 10.1.1.0 0.0.0.3
!
ip access-list extended OUTSIDE-IN
 permit udp any eq bootps any eq bootpc
 permit udp any eq domain any
 permit tcp any any established
 permit udp any any eq 51820
 permit icmp any any echo-reply
 permit icmp any any time-exceeded
 permit icmp any any unreachable
 deny ip any any log
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
 transport input telnet ssh
!
End
```

R2-Remote#show running-config

```
Building configuration...
Current configuration : 1698 bytes
!
version 15.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R2-Remote
!
ip cef
!
no ip domain lookup
!
ip dhcp excluded-address 172.16.0.1 172.16.0.9
!
ip dhcp pool CLIENT_LAN
 network 172.16.0.0 255.255.255.0
 default-router 172.16.0.1
 dns-server 8.8.8.8 8.8.4.4
!
interface FastEthernet0/0
 description WAN_TO_INTERNET_SW
 ip address dhcp
 ip access-group OUTSIDE-IN in
 ip nat outside
 duplex auto
 speed auto
 no shutdown
!
interface FastEthernet1/0
 description CLIENT_LAN
 ip address 172.16.0.1 255.255.255.0
 ip nat inside
 duplex auto
 speed auto
 no shutdown
!
access-list 20 permit 172.16.0.0 0.0.0.255
!
ip nat inside source list 20 interface FastEthernet0/0 overload
!
ip access-list extended OUTSIDE-IN
 permit udp any eq bootps any eq bootpc
 permit udp any eq domain any
 permit udp any eq 51820 any
 permit tcp any any established
 permit icmp any any echo-reply
 permit icmp any any time-exceeded
 permit icmp any any unreachable
 deny ip any any log
!
```

```
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
 transport input telnet ssh
!
end
```

Internet-SW#show running-config

```
Building configuration...

Current configuration : 1184 bytes
!
version 15.2
no service password-encryption
!
hostname Internet-SW
!
spanning-tree mode pvst
!
interface Ethernet0/0
 description TO_NAT_CLOUD
 switchport mode access
 spanning-tree portfast
 no shutdown
!
interface Ethernet0/1
 description TO_R1_CORPORATE
 switchport mode access
 spanning-tree portfast
 no shutdown
!
interface Ethernet0/2
 description TO_R2_REMOTE
 switchport mode access
 spanning-tree portfast
 no shutdown
!
interface Vlan1
 ip address 10.0.3.100 255.255.255.0
 no shutdown
!
ip default-gateway 10.0.3.2
!
line con 0
 logging synchronous
line vty 0 4
 login
!
End
```

LAN-SW#show running-config

```
Building configuration...
Current configuration : 1362 bytes
!
version 15.2
no service password-encryption
!
hostname LAN-SW
!
spanning-tree mode pvst
!
interface Ethernet0/0
 description TO_PFSENSE_LAN
 switchport mode access
 spanning-tree portfast
 no shutdown
!
interface Ethernet0/1
 description TO_FIREZONE
 switchport mode access
 spanning-tree portfast
 no shutdown
!
interface Ethernet0/2
 description TO_KEYCLOAK
 switchport mode access
 spanning-tree portfast
 no shutdown
!
interface Ethernet0/3
 description TO_INTERNAL_SERVER
 switchport mode access
 spanning-tree portfast
 no shutdown
!
interface Vlan1
 ip address 10.0.0.100 255.255.255.0
 no shutdown
!
ip default-gateway 10.0.0.1
!
line con 0
 logging synchronous
line vty 0 4
 login
!
end
```