ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «МЕХАНІЗМИ ЗАХИСТУ ІОТ-ПРИСТРОЇВ У КОРПОРАТИВНІЙ ІНФРАСТРУКТУРІ»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 Комп'ютерна інженерія
(код, найменування спеціальності)
освітньо-професійної програми «Комп'ютерна інженерія»
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ *Ілля ЛАБЗА*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. КІД-41
Ілля ЛАБЗА
Ім'я, ПРІЗВИЩЕ

Керівник: Артем АНТОНЕНКО
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Рецензент: _____
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерної інженерії

Ступінь вищої освіти бакалавр

Спеціальність 123 Комп'ютерна інженерія

Освітньо-професійна програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
_____ Наталія ЛАЩЕВСЬКА
«_____» _____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Лабза Ілля Григорович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Механізми захисту IoT-пристроїв у корпоративній інфраструктурі»
керівник кваліфікаційної роботи: Артем АНТОНЕНКО, кандидат технічних наук, доцент ,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «20» лютого 2026р. №51
2. Строк подання кваліфікаційної роботи «14» травня 2026р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, моделі та принципи сегментації мереж (VLAN, L3-маршрутизація), механізми міжмережевого екранування (ACL, firewall), програмні засоби віртуалізації (VMware Workstation), технології контейнеризації (Docker, macvlan), методи моделювання та тестування мережевих інфраструктур, підходи до аналізу та оцінки ефективності систем захисту.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   1. Теоретичні засади безпеки інфраструктури підключених пристроїв
   2. Проєктування та практична реалізація захищеного сегменту
   3. Оптимізація, порівняння та перспективи розвитку
5. Перелік ілюстративного матеріалу: *презентація*
   1. Мета, об'єкт, предмет дослідження
   2. Актуальність роботи
   3. Архітектура запропонованого рішення
   4. Схема адресації та сегментації
   5. Практична реалізація середовища
   6. Алгоритми фільтрації та безпеки
   7. Тестування та підтвердження ізоляції
   8. Оптимізація та продуктивність
   9. Порівняння підходів

10. Перспективи розвитку
11. Висновки
12. Публікації та апробація роботи

6. Дата видачі завдання «20» лютого 2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	22.02-20.03.2026	Виконано
2	Обробка матеріалу	20.03-30.03.2026	Виконано
3	Розробка теоретичної частини	31.03-10.04.2026	Виконано
4	Проєктування та практична реалізація захищеного сегменту	11.04-20.04.2026	Виконано
5	Висновки за результатами аналізу	21.04-26.04.2026	Виконано
6	Оформлення пояснювальної записки	26.04-28.04.2026	Виконано
7	Розробка презентації	29.04-01.05.2026	Виконано
8	Передзахист	08.05-11.05.2026	Виконано
9	Здача в деканат	08.05-14.05.2026	Виконано

Здобувач вищої освіти _____ Ілля ЛАБЗА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Артем АНТОНЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 64 стор., 30 рис., 2 табл., 70 джерел.

Мета роботи – мінімізація ризиків несанкціонованого доступу та поширення загроз у корпоративній мережі за рахунок використання принципів сегментації трафіку, міжмережевого екранування та ізоляції IoT-пристроїв.

Об'єкт дослідження – процеси функціонування та взаємодії IoT-пристроїв у корпоративній мережевій інфраструктурі.

Предмет дослідження – засоби та технології побудови захищених сегментів мережі, зокрема механізми VLAN-сегментації, маршрутизації, ACL та міжмережевих екранів.

Короткий зміст роботи: У роботі досліджено теоретичні аспекти безпеки IoT, методи мережевої ізоляції на основі VLAN та управління доступом через ACL. Розглянуто принципи сегментації трафіку та моделювання мереж у віртуальних середовищах із використанням гіпервізорів і технологій контейнеризації.

Практичну частину присвячено реалізації захищеного сегмента на базі програмного маршрутизатора pfSense та імітації IoT-пристроїв у контейнерах Docker. Налаштовано політики безпеки, проведено тестування ізоляції та оптимізовано роботу міжмережевого екрана з визначенням перспектив інтеграції систем IDS/IPS.

КЛЮЧОВІ СЛОВА: IOT, МЕРЕЖЕВА БЕЗПЕКА, СЕГМЕНТАЦІЯ, VLAN, МІЖМЕРЕЖЕВИЙ ЕКРАН, ACL, PFSENSE, DOCKER, ВІРТУАЛІЗАЦІЯ, ІЗОЛЯЦІЯ, IDS/IPS, КОРПОРАТИВНА МЕРЕЖА.

ЗМІСТ

ВСТУП 8
1. ТЕОРЕТИЧНІ ЗАСАДИ БЕЗПЕКИ ІНФРАСТРУКТУРИ ПІДКЛЮЧЕНИХ ПРИСТРОЇВ 9
1.1 Аналіз вразливостей обладнання у корпоративному середовищі 9
1.2 Концепція мережевої ізоляції та сегментації трафіку 13
1.3 Моделі управління доступом та міжмережеві екрани 17
1.4 Теоретичні основи моделювання мереж у віртуальних середовищах 22
1.5 Формування вимог до цільової системи захисту. 27
2. ПРОЄКТУВАННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАХИЩЕНОГО СЕГМЕНТУ 32
2.1 Архітектурна модель запропонованого рішення 32
2.2 Етапи побудови віртуального мережевого середовища 40
2.3 Алгоритми маршрутизації та конфігурація політик безпеки 48
2.4 Практична реалізація та тестування функціональності системи 53
3. ОПТИМІЗАЦІЯ, ПОРІВНЯННЯ ТА ПЕРСПЕКТИВИ РОЗВИТКУ 59
3.1 Методи оптимізації роботи мережевого екрану та усунення недоліків 59
3.2 Порівняльний аналіз із існуючими підходами до сегментації 63
3.3 Перспективи модернізації та масштабування системи безпеки 67
ВИСНОВКИ 71
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 72
Додаток А. Конфігураційний файл docker-compose.yml 78
Додаток Б. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація) 79

ВСТУП

Сучасні корпоративні системи зазнають змін через впровадження IoT-пристроїв у загальну інфраструктуру. Вони створюють нові загрози через обмежені обчислювальні ресурси, відсутність захисту та вразливі конфігурації. Теоретична частина роботи розглядає вектори атак, зокрема використання стандартних облікових даних, відкритих сервісів і недоліків прошивок, а також аналізує наслідки компрометації вузлів для всієї мережі.

Увага приділяється ризикам безпровідне підключення IoT: перехопленню трафіку, атакам «відмова в обслуговуванні» та порушенню сегментації. Обґрунтовано архітектурні підходи: логічне розділення VLAN та маршрутизацію третього рівня для контролю міжсегментного обміну. Такий метод ефективно обмежує поширення загроз та допомагає локалізувати потенційні інциденти безпеки.

Теоретичні засади доповнюються механізмами контролю: списками ACL та принципом мінімальних привілеїв («заборонено все, що не дозволено»). Розглядаються методи моделювання середовищ через гіпервізори й контейнеризацію для відтворення топологій та тестування захисту. Завершальний етап — формування вимог до цільової системи, враховуючи критерії ізоляції, метрики успішності та обмеження ресурсів.

Практична частина реалізує захищений сегмент у VMware Workstation із використанням pfSense, клієнтської машини та Docker для імітації IoT. Планується побудова сегментованої топології з окремими підмережами, налаштування політик маршрутизації та екранування. Тестування включає перевірку доступу до зовнішніх ресурсів і підтвердження повної ізоляції між сегментами.

1. ТЕОРЕТИЧНІ ЗАСАДИ БЕЗПЕКИ ІНФРАСТРУКТУРИ ПІДКЛЮЧЕНИХ ПРИСТРОЇВ

1.1 Аналіз вразливостей обладнання у корпоративному середовищі

Корпоративні інформаційні системи перебувають у стані інтенсивної трансформації, що зумовлена активним впровадженням концепції Інтернету речей, у межах якої до складу інфраструктури підприємств дедалі частіше інтегруються різноманітні пристрої, зокрема сенсори, камери спостереження, системи контролю доступу та інші елементи автоматизації, що забезпечують збір, обробку та передачу даних, водночас така інтеграція формує новий клас загроз для інформаційної безпеки, оскільки на відміну від традиційних серверів і робочих станцій подібні пристрої зазвичай не проєктуються з урахуванням жорстких вимог захисту, а орієнтація виробників на зниження вартості, енергоспоживання та підвищення продуктивності призводить до обмеженої реалізації механізмів безпеки, що робить такі вузли потенційними точками входу до корпоративної мережі та потребує детального аналізу можливих векторів атак і ризиків, пов’язаних із їх підключенням, зокрема через безпровідні середовища [1].

Вектори атак та наслідки компрометації периметра. Атаки на IoT-пристрої реалізуються через широкий спектр каналів і технічних вразливостей, серед яких вагоме місце займає експлуатація недоліків прошивки, що часто залишається застарілою або непідтримуваною, унаслідок чого виникають можливості виконання довільного коду, отримання привілейованого доступу та впровадження прихованих механізмів керування, паралельно із цим поширеними є атаки, пов’язані з аутентифікацією, де використання стандартних облікових даних на кшталт admin/admin, слабких паролів або відсутність механізмів обмеження кількості спроб входу значно спрощує несанкціонований доступ, додаткову небезпеку створює наявність відкритих мережевих портів і сервісів, таких як Telnet,

HTTP, FTP або незахищений MQTT, що дозволяє здійснювати сканування мережі, встановлювати незашифровані з’єднання та перехоплювати дані, при цьому відсутність або некоректна реалізація криптографічного захисту сприяє реалізації атак типу Man-in-the-Middle, у межах яких можливе перехоплення та модифікація переданих пакетів, а також підміна інформації [2].

Окрему категорію становлять масові атаки із залученням ботнетів, де скомпрометовані IoT-пристрої використовуються як елементи розподіленої інфраструктури для здійснення DDoS-атак, розповсюдження шкідливого програмного забезпечення та подальшого сканування мереж, однак ключова проблема полягає не лише у наявності вразливостей як таких, а у топології розміщення пристроїв, оскільки їх інтеграція в один сегмент із корпоративними серверами, базами даних і робочими станціями створює умови для горизонтального переміщення зловмисника, де початковий доступ до одного пристрою, наприклад камери спостереження, може бути використаний як проміжна точка для проникнення у внутрішню мережу, що надалі забезпечує доступ до критичних ресурсів, включаючи сервери баз даних, при цьому типовий сценарій компрометації розвивається послідовно через етапи отримання первинного доступу шляхом підбору облікових даних або використання експлойтів, закріплення у системі через встановлення бекдорів і зміну конфігурації, проведення розвідки з аналізом IP-діапазонів і визначенням активних вузлів, подальшого латерального переміщення із використанням довірених з’єднань і завершальної фази, у межах якої здійснюється компрометація критичних ресурсів (див. рис. 1.1).



Рисунок 1.1 – Сценарій проникнення через скомпрометований IoT-вузол

Наслідки таких інцидентів мають комплексний характер і проявляються у:

— порушенні конфіденційності через витік комерційної інформації;

— втраті цілісності внаслідок несанкціонованої модифікації даних або команд керування;

— порушенні доступності сервісів через перевантаження або реалізацію атак типу DDoS.

Все це в сукупності призводить до фінансових втрат, простоїв у роботі інфраструктури та репутаційних ризиків, що підкреслює необхідність системного підходу до побудови захищеної мережевої архітектури із врахуванням особливостей IoT-пристроїв та їх ролі у корпоративному середовищі [3].

Проблематика загальнодоступних безпровідних мереж. Безпровідні мережі, зокрема Wi-Fi у діапазоні 2,4 ГГц, широко застосовуються для підключення IoT-пристроїв завдяки простоті розгортання, низькій вартості впровадження та відсутності необхідності у кабельній інфраструктурі, що суттєво спрощує інтеграцію пристроїв у корпоративне середовище, однак така доступність супроводжується підвищеним рівнем ризиків, пов’язаних із відкритою природою радіоканалу передачі даних, у межах якого будь-який пристрій у зоні покриття потенційно здатен здійснювати аналіз трафіку або впливати на процес обміну інформацією, що створює передумови для реалізації атак, пов’язаних із перехопленням і модифікацією даних.

Одним із найбільш поширених сценаріїв є атака типу Man-in-the-Middle, за якої зловмисник створює підроблену точку доступу з параметрами, ідентичними до легітимної мережі, при цьому більш потужний сигнал змушує IoT-пристрій автоматично підключитися саме до неї, унаслідок чого весь мережевий трафік проходить через контрольований вузол, де може бути перехоплений, проаналізований або змінений під час передачі, паралельно із цим можливе застосування атак глушіння сигналу, що полягають у створенні радіоперешкод у відповідному діапазоні частот, унаслідок чого порушується стабільність роботи мережі, виникають збої у передачі даних та втрата зв’язку між пристроями, що може використовуватися як допоміжний механізм для примусового

перепідключення до підроблених точок доступу або для дестабілізації функціонування інфраструктури (див. рис. 1.2) [4].

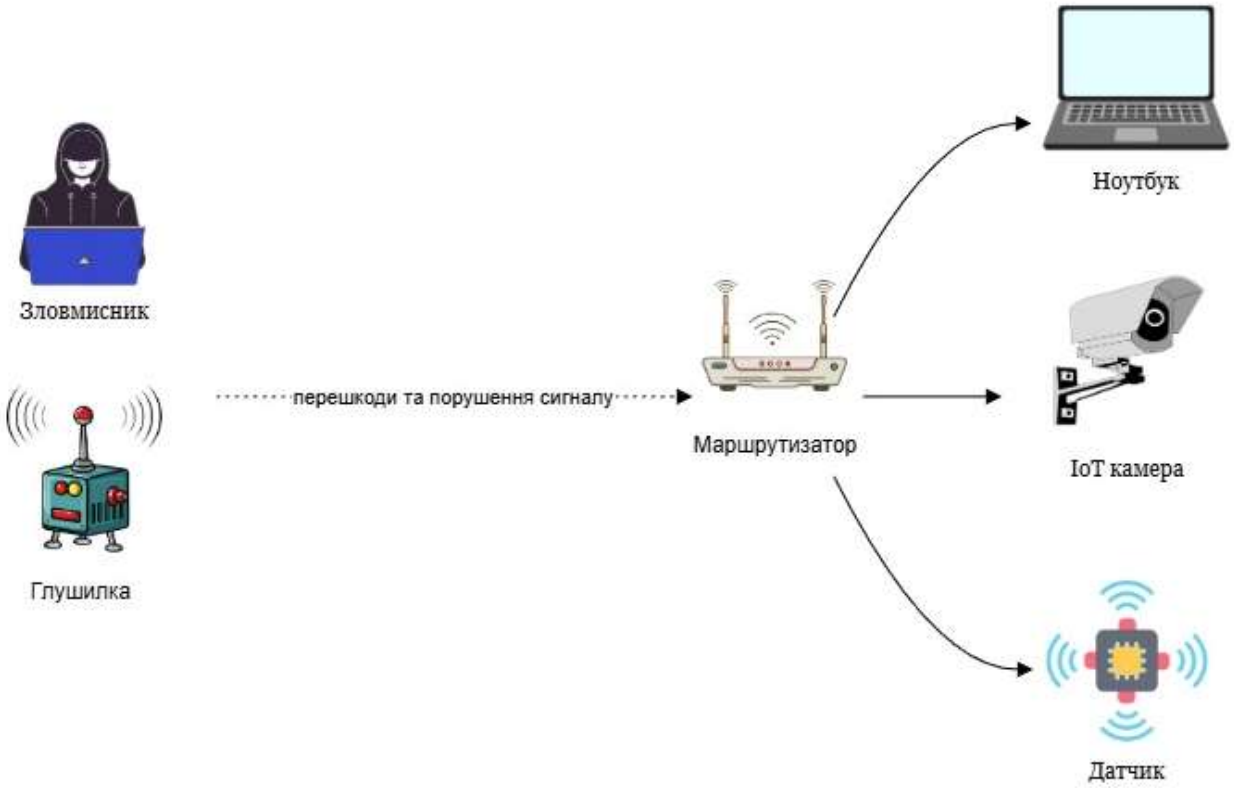


Рисунок 1.2 – Вплив jamming-атаки на безпровідне середовище

Додаткову загрозу становить несанкціонований доступ до мережі, який стає можливим у разі використання слабких параметрів безпеки, таких як застосування стандартів WPA або WPA2 без складного пароля, відсутність фільтрації MAC-адрес чи інших механізмів контролю доступу, унаслідок чого сторонній користувач може безперешкодно підключитися до корпоративної мережі та отримати доступ до внутрішніх ресурсів, що ускладнюється ще й тим, що у багатьох випадках використовується єдина безпровідна мережа для обслуговування як корпоративних користувачів, так і IoT-пристроїв, у результаті чого всі вузли опиняються в одному логічному сегменті без належної ізоляції [5].

Така архітектура створює умови для швидкого розвитку атак, оскільки зловмисник після підключення отримує можливість виконувати сканування всієї мережі, виявляти активні вузли, поширювати шкідливе програмне забезпечення,

компрометувати облікові записи та отримувати доступ до внутрішніх сервісів, які не повинні бути доступними ззовні, проведений аналіз підтверджує, що IoT-пристрої у поєднанні з незахищеним або неправильно організованим безпровідним середовищем формують один із найбільш уразливих сегментів корпоративної інфраструктури через обмежені можливості вбудованого захисту, наявність типових вразливостей, некоректне мережеве розміщення та відсутність сегментації, причому визначальним фактором ризику є не лише сам пристрій, а спосіб його інтеграції у мережу без механізмів контролю доступу, що обґрунтовує необхідність застосування підходів мережевої ізоляції, сегментації трафіку та використання міжмережевих екранів як базових засобів забезпечення безпеки.

1.2 Концепція мережевої ізоляції та сегментації трафіку

Концепція мережевої ізоляції та сегментації трафіку формується як відповідь на виявлену раніше проблему, пов’язану не лише з вразливістю окремих IoT-пристроїв, а й із їх неконтрольованою інтеграцією у спільний мережевий простір, у якому відсутні чіткі межі доступу та політики взаємодії між різними класами вузлів, у таких умовах навіть один скомпрометований елемент інфраструктури здатний стати відправною точкою для масштабованої атаки, що поширюється на інші сегменти системи через механізми довірених з’єднань та внутрішньої маршрутизації, тому основною метою сегментації виступає обмеження області поширення загрози, тобто зменшення так званого радіусу ураження, а також локалізація потенційних інцидентів із можливістю їх контролю та швидкого реагування, при цьому ізоляція виступає як більш жорсткий механізм, що обмежує або повністю виключає взаємодію між окремими зонами, формуючи чітко визначені політики доступу між ними та дозволяючи забезпечити контрольований обмін даними лише у дозволених напрямках.

У контексті корпоративної інфраструктури із використанням IoT-пристроїв така модель дозволяє розмежувати вузли з різним рівнем довіри, де сенсори, камери та інші периферійні елементи, які мають обмежені механізми захисту, ізолюються від критичних ресурсів, таких як сервери баз даних або робочі станції користувачів, у результаті чого навіть у разі компрометації одного з елементів атака не може безпосередньо поширитися на інші частини мережі без подолання додаткових рівнів контролю, що суттєво ускладнює реалізацію сценаріїв латерального переміщення та знижує ризик повномасштабного інциденту безпеки.

Принципи логічного розділення середовищ (VLAN). Реалізація логічного розділення мережі досягається за допомогою технології віртуальних локальних мереж, які дозволяють поділити одну фізичну мережеву інфраструктуру на декілька незалежних широкомовних доменів на канальному рівні моделі OSI, у такому підході кожен сегмент отримує власний ідентифікатор, а мережевий трафік маркується відповідно до стандарту IEEE 802.1Q шляхом додавання спеціального тегу до Ethernet-кадрів, що містить інформацію про належність пакета до конкретного сегмента, у результаті комутатори здатні розділяти потоки даних і передавати їх лише в межах визначеного домену, запобігаючи несанкціонованому обміну між різними групами пристроїв, при цьому порти доступу використовуються для підключення кінцевих вузлів і працюють в межах одного сегмента без тегування, тоді як транкові порти забезпечують передачу трафіку декількох сегментів між мережевими пристроями із використанням тегування, що дозволяє будувати масштабовані та гнучкі топології без необхідності фізичного розділення інфраструктури (див. рис. 1.3) [6].

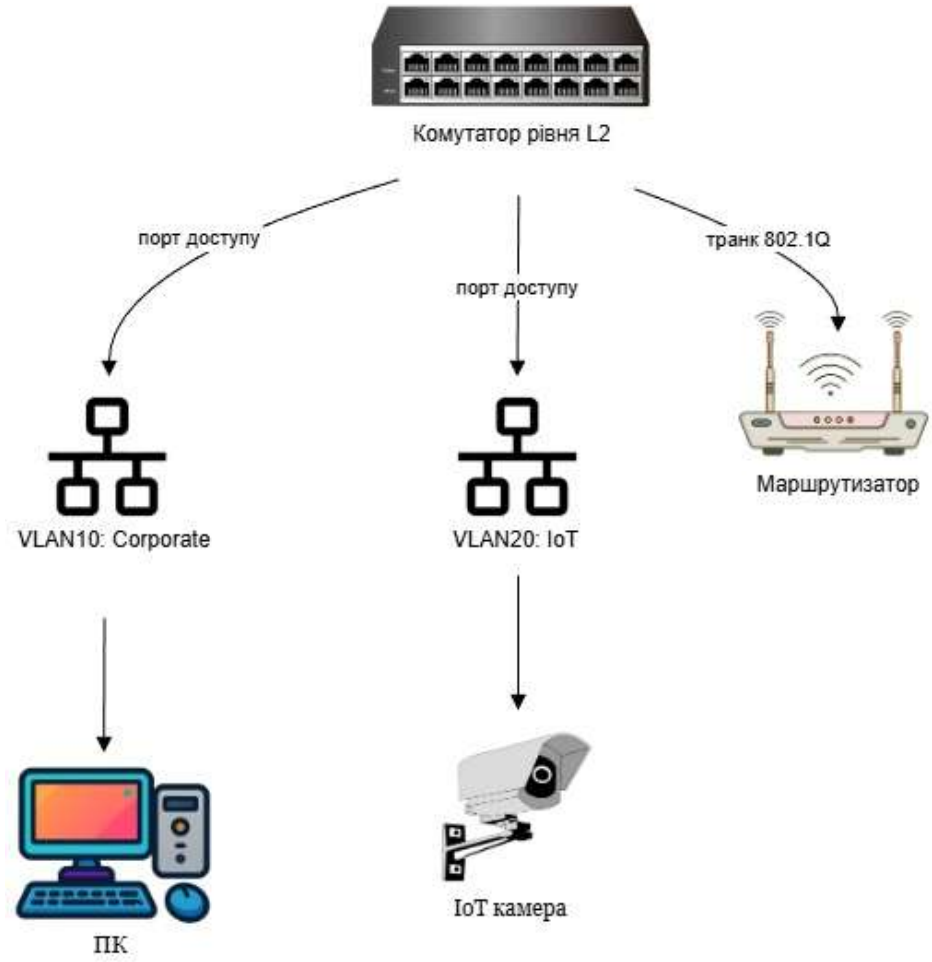


Рисунок 1.3 – Принцип роботи VLAN у комутованій мережі

Практичне застосування такого підходу передбачає створення окремих логічних сегментів для різних типів пристроїв, наприклад виділення корпоративного сегмента для робочих станцій, окремого сегмента для IoT-пристроїв та ізольованого сегмента для серверної частини, при цьому взаємодія між цими сегментами не здійснюється безпосередньо, а проходить через маршрутизатор рівня L3, який виконує функцію контролю доступу та реалізує політики безпеки, що дозволяє не лише ізолювати трафік, але й детально регламентувати дозволені напрямки обміну даними, зменшувати обсяг широкомовного трафіку, підвищувати загальну продуктивність мережі та знижувати поверхню атаки, водночас слід враховувати, що сама по собі сегментація на рівні VLAN не забезпечує повного захисту через відсутність вбудованого механізму контролю між сегментами, можливість реалізації

специфічних атак та необхідність застосування додаткових засобів маршрутизації і фільтрації трафіку, що в комплексі формує багаторівневу модель безпеки, орієнтовану на ізоляцію, контроль та мінімізацію ризиків [7].

Роль маршрутизації третього рівня у контролі обміну. Оскільки сегментація мережі за допомогою віртуальних локальних мереж забезпечує ізоляцію на канальному рівні, будь-яка взаємодія між окремими сегментами можлива виключно через пристрій третього рівня, яким виступає маршрутизатор або комутатор рівня L3, при цьому саме цей елемент інфраструктури стає центральною точкою контролю трафіку та реалізації політик безпеки, через яку проходять усі міжсегментні потоки даних, що дозволяє не лише фізично розмежувати трафік, але й логічно контролювати дозволені напрямки обміну, формуючи керовану модель доступу між різними зонами мережі.

Принцип функціонування такого централізованого контролю полягає в обов’язковій перевірці кожного пакета, який намагається перетнути межу між сегментами, при цьому маршрутизатор аналізує ключові параметри трафіку, включаючи адресу джерела, адресу призначення та тип протоколу, і приймає рішення на основі попередньо визначених правил, що реалізують модель «дозволено лише те, що явно визначено», у результаті чого будь-який трафік, який не відповідає заданим політикам, автоматично відхиляється без можливості подальшого проходження, а дозволені з’єднання чітко обмежуються встановленими умовами, що суттєво підвищує контрольованість мережевої взаємодії [8].

У контексті використання IoT-пристроїв така модель дозволяє реалізувати сувору політику доступу, за якої сегмент із пристроями з низьким рівнем довіри ізолюється від корпоративної мережі та зовнішніх ресурсів, при цьому взаємодія обмежується лише необхідними сервісами, наприклад доступом до спеціалізованого сервера збору даних або брокера повідомлень, тоді як прямий обмін із робочими станціями або вихід у глобальну мережу блокується, що унеможливлює використання скомпрометованих пристроїв як проміжної ланки для подальшого поширення атаки або витоку інформації, а вся логіка взаємодії

концентрується в єдиній точці контролю, де може бути реалізовано аудит, логування та додатковий аналіз трафіку (див. рис. 1.4) [9].

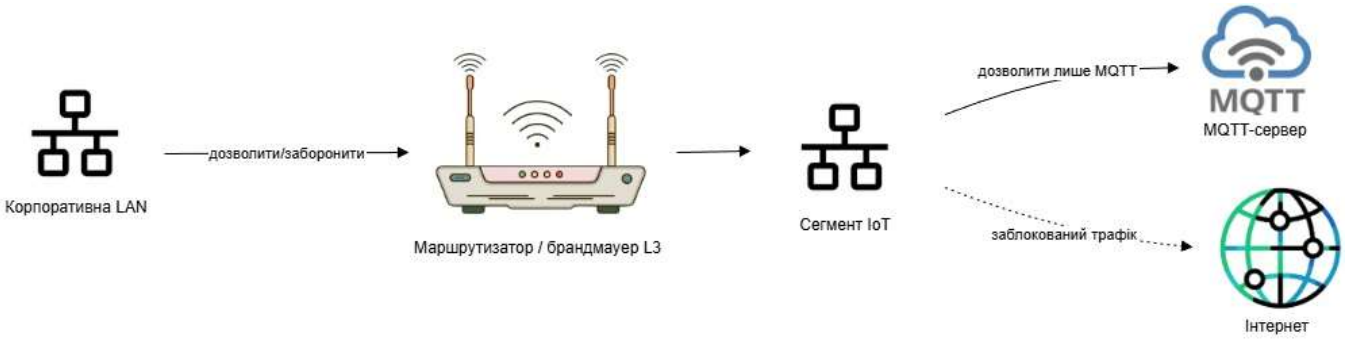


Рисунок 1.4 – Схема маршрутизації між сегментами

Поєднання сегментації на рівні VLAN та маршрутизації третього рівня формує цілісну архітектуру безпеки, у якій кожен сегмент функціонує як окрема зона з визначеними правилами доступу, а міжсегментна взаємодія відбувається виключно під контролем маршрутизатора, що дозволяє ефективно запобігати несанкціонованому доступу, обмежувати можливості горизонтального переміщення всередині мережі та створювати основу для впровадження більш складних механізмів фільтрації, контролю доступу та міжмережевого екранування, які забезпечують багаторівневий підхід до захисту корпоративної інфраструктури.

1.3 Моделі управління доступом та міжмережеві екрани

Ефективна сегментація мережі на основі логічного розділення та ізоляції на канальному рівні формує лише базову структуру безпеки, однак без впровадження механізмів контролю доступу така структура залишається вразливою до несанкціонованої взаємодії між сегментами, що зумовлює необхідність застосування інструментів, які функціонують на мережевому та транспортному рівнях моделі OSI і забезпечують керовану фільтрацію трафіку відповідно до

визначених політик, у цьому контексті міжмережеві екрани виступають як окремі або інтегровані компоненти інфраструктури, що реалізують логіку прийняття рішень щодо дозволу або блокування пакетів, формуючи контрольоване середовище обміну даними між різними зонами мережі, де кожен потік аналізується перед передачею і підпорядковується заданим правилам безпеки, що дозволяє [10]:

— обмежити доступ до критичних ресурсів;

— зменшити ризик несанкціонованих підключень;

— забезпечити прогнозовану поведінку мережевої взаємодії.

Використання списків контролю доступу (ACL). Списки контролю доступу є одним із базових механізмів реалізації такої фільтрації і представляють собою впорядковану сукупність правил, кожне з яких визначає умови обробки мережевого пакета на основі його атрибутів, зокрема:

— адреси джерела;

— адреси призначення;

— типу протоколу;

— номерів портів.

При цьому кожне правило містить відповідну дію, що визначає подальшу долю пакета, а саме дозвіл або блокування, функціонування такого механізму ґрунтується на послідовній обробці правил зверху вниз, де для кожного пакета здійснюється перевірка відповідності першому знайденому правилу, після чого виконується визначена дія і подальша перевірка припиняється, у разі відсутності збігу застосовується неявне правило заборони, яке забезпечує відхилення будь-якого неописаного трафіку, що формує модель суворого контролю доступу і дозволяє реалізувати принцип мінімальних привілеїв у мережевому середовищі (див. рис. 1.5) [11].

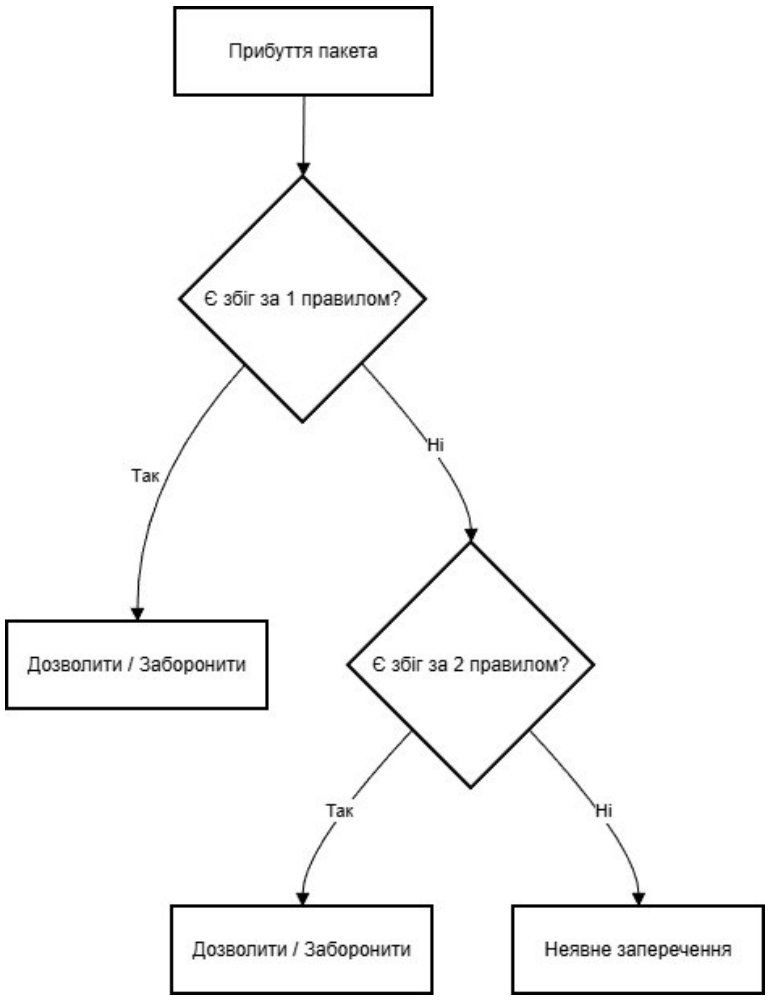


Рисунок 1.5 – Логіка обробки ACL-правил (First Match)

Розрізняють базові варіанти списків контролю доступу, що обмежуються аналізом лише адреси джерела, та розширені механізми, які враховують повний набір параметрів пакета, включаючи адресу призначення, тип протоколу та порт, що дозволяє реалізувати більш гнучкі та деталізовані політики безпеки, у практичній реалізації для корпоративного сегмента та IoT-інфраструктури формуються правила, які обмежують прямий доступ між сегментами, дозволяючи лише визначені типи трафіку, наприклад доступ до окремих сервісів або можливість виконання діагностичних запитів, при цьому вся взаємодія проходить через централізовану точку контролю, де здійснюється фільтрація та логування подій, що дозволяє відстежувати спроби несанкціонованого доступу та аналізувати поведінку мережевих вузлів.

Разом із перевагами такого підходу існують обмеження, пов’язані з відсутністю аналізу стану з’єднань, що характерно для статичних механізмів фільтрації, складністю масштабування при великій кількості правил, а також підвищеним ризиком помилок конфігурації, які можуть призвести до небажаного відкриття доступу або блокування легітимного трафіку, додатковим фактором виступає відсутність глибокого аналізу вмісту пакетів, що обмежує можливості виявлення складних атак, однак навіть з урахуванням цих недоліків використання списків контролю доступу у поєднанні з міжмережевими екранами формує фундамент для побудови контрольованого та ізольованого мережевого середовища, у якому кожна взаємодія регламентується чітко визначеними правилами та підпорядковується загальній політиці безпеки корпоративної інфраструктури [12].

Принцип мінімальних привілеїв. Принцип мінімальних привілеїв розглядається як одна з фундаментальних засад побудови безпечних інформаційних систем, у межах якої кожен суб’єкт мережевої інфраструктури отримує лише той набір прав і доступів, що є необхідним для виконання його функціонального призначення, у результаті чого будь-які надлишкові можливості взаємодії виключаються ще на етапі проєктування політик безпеки, що дозволяє суттєво знизити ризик несанкціонованого доступу та обмежити потенційні наслідки компрометації окремих вузлів, при цьому реалізація такого підходу у мережевому середовищі зазвичай базується на моделі «заборонено за замовчуванням», де будь-який трафік, який не визначений як дозволений, автоматично блокується, а дозволи формуються у вигляді чітко визначених винятків із загальної політики заборони.

У контексті використання IoT-пристроїв застосування цього принципу набуває особливої актуальності, оскільки такі пристрої зазвичай не потребують широкого спектра мережевих взаємодій і виконують обмежений набір функцій, пов’язаних із передачею даних до визначених сервісів, що дозволяє сформувати максимально жорстку політику доступу, за якої всі вхідні та вихідні з’єднання блокуються, окрім тих, що є критично необхідними для роботи, зокрема

синхронізація часу через протокол NTP, звернення до локального DNS-сервера для резолюції імен, передача даних до визначеного брокера повідомлень або доступ до внутрішніх сервісів через контрольовані інтерфейси, у результаті чого будь-які спроби встановлення додаткових з'єднань, включаючи прямий доступ до корпоративних ресурсів або вихід у зовнішню мережу, автоматично відхиляються [13].

Алгоритмічно така модель реалізується через перевірку кожного пакета, що ініціюється з IoT-сегмента, на відповідність списку явно дозволених правил, де у випадку наявності відповідності виконується дія пропуску, а за відсутності збігу пакет відкидається без подальшої обробки, що формує жорсткий контроль трафіку та виключає можливість несанкціонованої взаємодії, при цьому порівняння підходів до формування політик доступу демонструє суттєву різницю між моделлю «дозволено за замовчуванням», яка передбачає відкритість мережі та високий рівень ризику, і моделлю «заборонено за замовчуванням», яка забезпечує значно вищий рівень безпеки за рахунок обмеження доступу лише визначеними сценаріями використання (див. рис. 1.6).

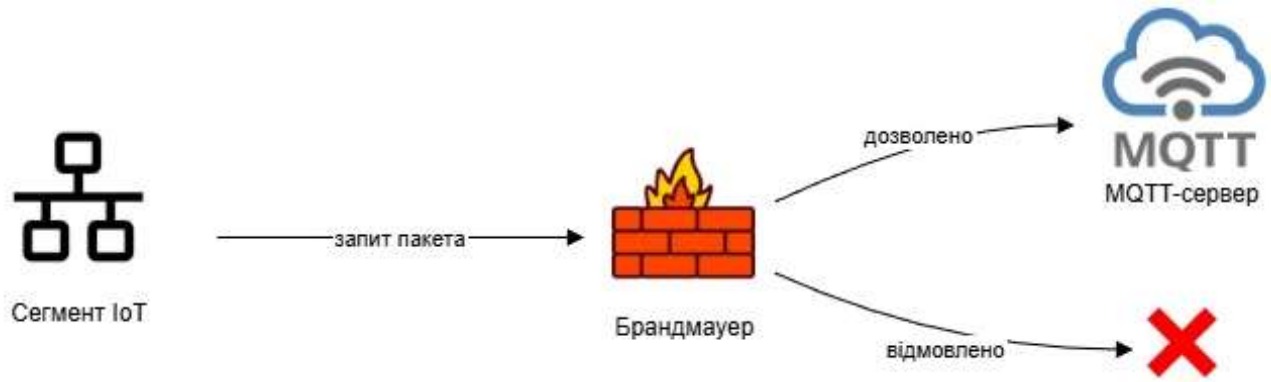


Рисунок 1.6 – Модель "Default Deny" у мережевій безпеці

Практичне застосування такого підходу передбачає ізоляцію IoT-сегмента від корпоративної мережі та зовнішнього середовища із дозволом взаємодії лише з конкретними сервісами, наприклад сервером обробки даних або брокером повідомлень, що унеможливлює використання пристроїв як проміжної ланки для поширення атак, а також дозволяє локалізувати потенційні інциденти в межах

одного сегмента, у результаті чого зменшується поверхня атаки, підвищується контрольованість мережевої взаємодії та забезпечується стабільність функціонування системи навіть у разі виникнення помилок або компрометації окремих компонентів, водночас ефективність реалізації залежить від коректності налаштування правил, уникнення надмірно широких дозволів та дотримання логічної послідовності обробки трафіку, що формує основу для подальшого переходу до практичного моделювання захищеного мережевого середовища.

1.4 Теоретичні основи моделювання мереж у віртуальних середовищах

Перед впровадженням будь-якого архітектурного рішення у реальному корпоративному середовищі доцільним є проведення попереднього тестування в контрольованих умовах, що дозволяє:

- перевірити коректність конфігурації;
- оцінити ефективність механізмів захисту;
- виявити потенційні недоліки без ризику для виробничої інфраструктури.

При цьому застосування засобів віртуалізації забезпечує можливість моделювання складних мережевих топологій із високим рівнем точності, оскільки використання гіпервізорів і контейнерних технологій дозволяє створювати ізольовані середовища, які відтворюють поведінку реальних мереж, включаючи:

- маршрутизацію;
- сегментацію;
- фільтрацію трафіку;
- взаємодію вузлів.

Все це формує основу для експериментального дослідження архітектурних підходів і перевірки сценаріїв атак та захисту.

Концепція віртуальних комутаторів. Віртуалізація мережі розглядається як процес створення логічних мережевих ресурсів, що функціонують незалежно від фізичної інфраструктури, де у середовищах типу VMware Workstation реалізація досягається через механізм віртуальних комутаторів, які забезпечують передачу кадрів між підключеними вузлами, ізоляцію трафіку та підтримку різних режимів роботи, серед яких bridged-режим із прямим підключенням до фізичної мережі та отриманням IP-адреси з тієї ж підмережі, NAT із трансляцією адрес і доступом до зовнішніх ресурсів через хост-систему, а також host-only режим, що формує повністю ізольовану мережу для тестування взаємодії між віртуальними машинами, при цьому концепція VMnet виступає аналогом VLAN у фізичних мережах, оскільки дозволяє створювати декілька незалежних логічних сегментів, кожен з яких функціонує як окремий широкомовний домен, у результаті чого підключення різних віртуальних машин до різних VMnet забезпечує повноцінну реалізацію мережевої сегментації у віртуальному середовищі (див. рис. 1.7) [14].

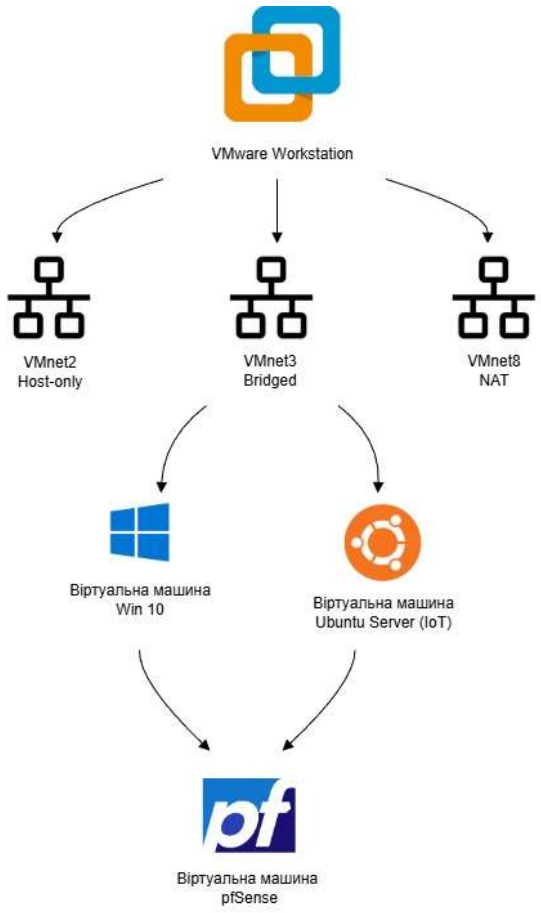


Рисунок 1.7 – Робота віртуальних мереж VMnet у VMware

Архітектура сегментованого віртуального рішення передбачає використання гіпервізора як базового середовища, у межах якого створюються декілька ізольованих мереж:

1. VMnet2 для корпоративного сегменту;
2. VMnet3 для IoT-середовища.

Тут кожен сегмент функціонує автономно без прямої взаємодії, а обмін даними здійснюється виключно через окрему віртуальну машину, що виконує функції маршрутизатора або міжмережевого екрану, у практичній реалізації таким елементом може виступати програмний маршрутизатор pfSense, який підключається до обох сегментів і забезпечує централізований контроль трафіку, фільтрацію пакетів та реалізацію політик безпеки, при цьому кожен сегмент залишається ізольованим, а будь-яка взаємодія проходить через контрольовану точку доступу, що дозволяє моделювати реальні сценарії функціонування корпоративної мережі з урахуванням вимог безпеки.

Застосування віртуальних комутаторів забезпечує:

1. моделювання фізичних підключень;
2. імітації мережевої топології;
3. проведення експериментів із різними сценаріями атак і захисту.

Все тому, що кожен VMnet відповідає окремому логічному сегменту, подібному до VLAN або фізичного комутатора, що дозволяє відтворити поведінку реальної мережі без використання апаратного обладнання, при цьому основними перевагами такого підходу є повна ізоляція середовища, гнучкість конфігурації, швидкість розгортання та можливість багаторазового тестування різних варіантів архітектури, тоді як обмеження проявляються у залежності від ресурсів хост-системи, можливих обмеженнях пропускної здатності та відсутності впливу фізичних факторів, таких як затримки сигналу або апаратні несправності, що необхідно враховувати під час інтерпретації результатів моделювання.

Імітація вузлів за допомогою контейнеризації. У класичному підході до моделювання мереж кожен вузол реалізується як окрема віртуальна машина, що призводить до значного споживання обчислювальних ресурсів, зокрема

оперативної пам’яті та процесорного часу, унаслідок чого масштабування середовища стає обмеженим і менш ефективним, особливо при необхідності імітації великої кількості IoT-пристроїв, при цьому альтернативою виступає контейнеризація, яка передбачає віртуалізацію на рівні операційної системи та дозволяє створювати ізольовані середовища у вигляді контейнерів, що використовують спільне ядро, завдяки чому забезпечується зменшення навантаження на ресурси, швидкий запуск та достатній рівень ізоляції для мережевого моделювання, а застосування платформи Docker як одного з найпоширеніших інструментів контейнеризації дозволяє автоматизувати процес розгортання, налаштування та управління контейнеризованими середовищами з підтримкою мережевої взаємодії між вузлами [15].

Особливу увагу приділено мережевим режимам контейнерів, серед яких критичним для моделювання IoT-середовища виступає режим "macvlan", що забезпечує можливість призначення кожному контейнеру унікальної MAC-адреси та інтеграції у віртуальну або фізичну мережу як повноцінного мережевого вузла, у результаті чого контейнери можуть отримувати власні IP-адреси та взаємодіяти з іншими сегментами через маршрутизатор, формуючи реалістичну модель мережевої інфраструктури, наприклад у межах одного сервера може бути розгорнуто декілька контейнерів із різними IP-адресами, підключених до окремого сегменту типу VMnet, де кожен контейнер імітує окремий IoT-пристрій, а взаємодія між сегментами здійснюється через програмний маршрутизатор, що дозволяє реалізувати централізований контроль доступу та фільтрацію трафіку (див. рис. 1.8) [16].

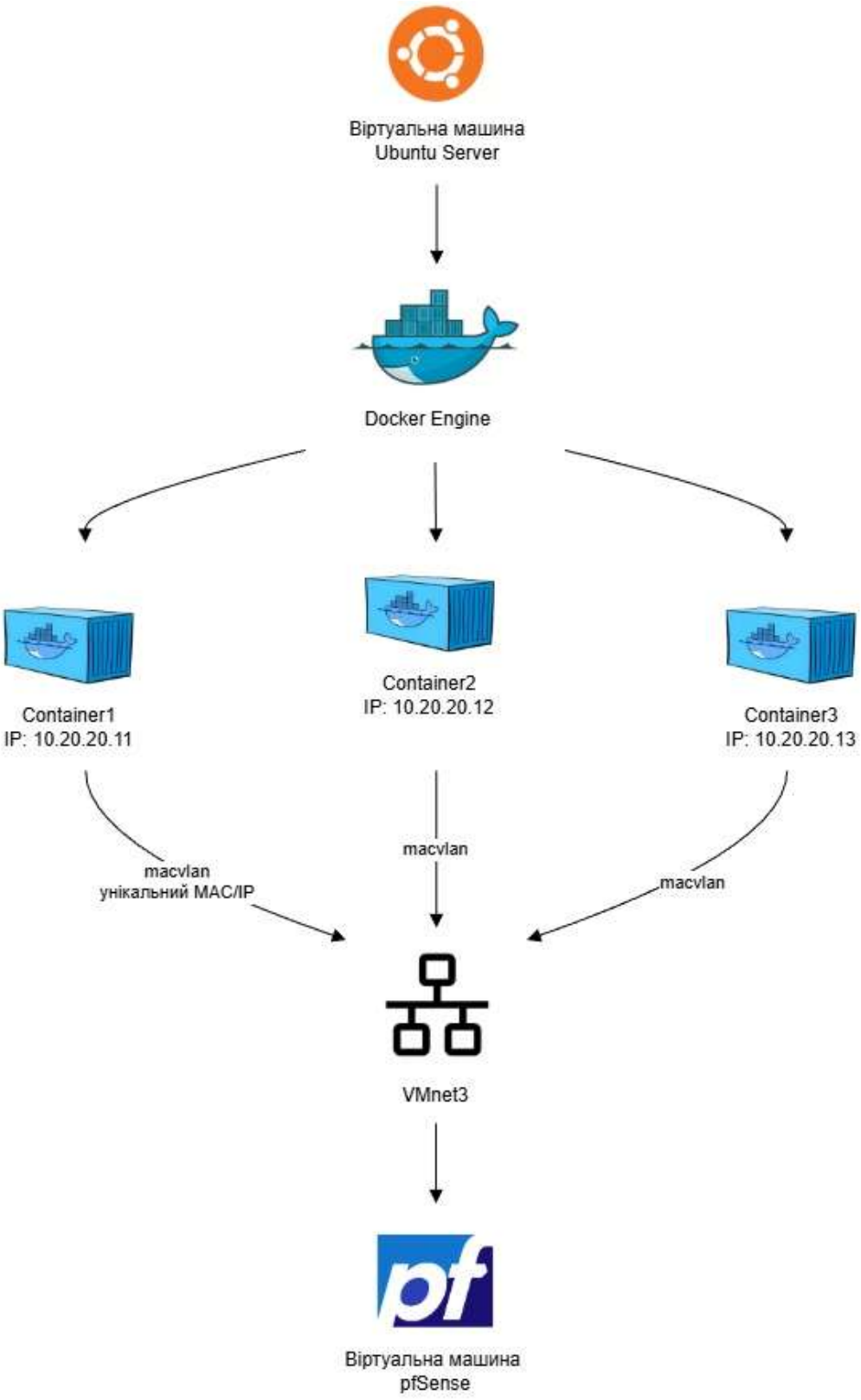


Рисунок 1.8 – Принцип роботи мережі "macvlan"

Порівняння контейнерів із віртуальними машинами демонструє суттєві відмінності у використанні ресурсів та швидкості розгортання, оскільки віртуальні

машини характеризуються високими витратами ресурсів і повільним запуском, тоді як контейнери є легковаговими, швидко ініціалізуються та забезпечують високу масштабованість, хоча рівень ізоляції у контейнерах є логічним і поступається повній ізоляції віртуальних машин, що необхідно враховувати під час побудови тестового середовища, при цьому процес розгортання контейнеризованої інфраструктури передбачає створення мережі, визначення конфігурації контейнерів, призначення мережевих параметрів, запуск екземплярів та перевірку їхньої взаємодії, що дозволяє швидко відтворювати різні сценарії функціонування мережі.

Застосування контейнеризації у поєднанні з віртуальними мережами забезпечує ефективне моделювання корпоративних інфраструктур із великою кількістю IoT-пристроїв, дозволяючи досліджувати їхню взаємодію, тестувати політики безпеки та оцінювати вплив різних архітектурних рішень, при цьому контейнери можуть бути інтегровані в окремі сегменти мережі та підпорядковуватися правилам доступу, визначеним на рівні маршрутизатора або міжмережевого екрану, що забезпечує відповідність моделі реальним умовам експлуатації, а поєднання віртуальних комутаторів і контейнеризації формує гнучке тестове середовище, здатне відтворювати складні сценарії, аналізувати ефективність механізмів захисту та оптимізувати архітектуру мережі перед її впровадженням у виробниче середовище.

1.5 Формування вимог до цільової системи захисту

Після аналізу загроз, розгляду принципів сегментації, механізмів контролю доступу та можливостей моделювання виникає необхідність формалізації вимог до цільової системи захисту IoT-пристроїв у корпоративному середовищі, оскільки чітке визначення таких вимог забезпечує [17]:

1. перехід від теоретичних положень до конкретних технічних рішень;

2. задає критерії оцінки ефективності;

3. створює основу для об’єктивного тестування і валідації результатів.

При цьому формування вимог передбачає визначення показників надійності, метрик успішності та встановлення обмежень щодо апаратно-програмного середовища, необхідного для реалізації експериментального стенду, що дозволяє забезпечити узгодженість архітектури із поставленими цілями та умовами експлуатації.

Критерії надійності та метрики успішності. Ефективність системи захисту визначається не лише функціональними можливостями, а здатністю протидіяти конкретним загрозам, ідентифікованим раніше, у зв’язку з чим основною ціллю виступає забезпечення контрольованої ізоляції IoT-сегменту та запобігання несанкціонованим взаємодіям між мережами, при цьому критерії оцінки мають бути вимірюваними, відтворюваними та придатними до експериментальної перевірки, серед яких ключовим виступає повна мережева ізоляція, що передбачає відсутність доступу з корпоративного сегменту до IoT-пристроїв, неможливість виконання ICMP-запитів, встановлення TCP-з’єднань або доступу до сервісів, аналогічно має бути відсутній будь-який ініційований доступ з IoT-сегменту до корпоративної мережі та заблокований вихідний трафік IoT до зовнішніх ресурсів, що формує замкнене середовище з жорстко контрольованими точками взаємодії.

Формалізована модель перевірки передбачає послідовну валідацію умов ізоляції шляхом ініціації тестових запитів, де успішне проходження визначається відсутністю відповіді на ICMP-запити, неможливістю встановлення TCP-з’єднань та відсутністю відкритих маршрутів, що підтверджує коректність налаштувань, при цьому додатковими вимогами виступають контрольованість трафіку, яка забезпечується через явне визначення дозволених з’єднань, прозорість і можливість аудиту через ведення журналів міжмережевого екрану та аналіз заблокованих пакетів, а також стійкість до помилок конфігурації, що досягається застосуванням принципу заборони за замовчуванням та мінімізації кількості правил, у результаті чого система набуває передбачуваної та контрольованої поведінки [18].

Практична перевірка реалізується за допомогою інструментів тестування мережевої взаємодії, таких як ICMP-запити, спроби встановлення TCP-з’єднань через утиліти типу "telnet" або "curl", сканування портів та аналіз журналів, де результати у вигляді відсутності відповіді або тайм-аутів підтверджують досягнення необхідного рівня ізоляції, а критерії надійності у такому підході базуються на поєднанні суворої сегментації, контрольованого доступу та можливості експериментальної верифікації, що формує цілісну основу для оцінки ефективності системи захисту та її подальшого впровадження у корпоративну інфраструктуру.

Вимоги до апаратно-програмних ресурсів. Для фіналізації та практичної реалізації моделі захищеної мережі застосовується спеціалізований віртуальний прототипний стенд, який розгортається в межах однієї фізичної хост-системи, що дозволяє [19]:

1. мінімізувати фінансові витрати;
2. забезпечити повну відтворюваність експериментів;
3. суттєво спростити процес тестування і налагодження всієї інфраструктури.

Причому така архітектура передбачає централізоване розміщення всіх компонентів у межах єдиного програмного середовища з чітким розподілом ролей між віртуальними вузлами. Базове середовище реалізації включає операційну систему хоста Windows 11, у межах якої функціонує гіпервізор VMware Workstation, що виступає платформою для створення та керування віртуальними машинами, серед яких окремо виділяється віртуальний маршрутизатор на базі pfSense, клієнтська система на базі Windows 10, а також серверна система на базі Ubuntu Server із встановленим Docker Engine, який забезпечує контейнеризацію та імітацію IoT-пристроїв у вигляді ізольованих процесів із власними мережевими параметрами. У межах цієї взаємодії хост-система виступає базовим рівнем, всередині якого через механізми віртуалізації створюється три логічно відокремлені вузли:

1. VM1 виконує функції маршрутизатора та міжмережевого екрану;
2. VM2 представляє клієнтський сегмент корпоративної мережі;

3. VM3 реалізує IoT-сегмент із контейнеризованими пристроями, що взаємодіють через окремий віртуальний комутатор VMnet3;

4. клієнтська система підключена до VMnet2.

Все це забезпечує чітке логічне розділення трафіку.

Стабільність функціонування такого середовища напряму залежить від апаратних характеристик хост-системи, де мінімально допустимий обсяг оперативної пам’яті становить 8 ГБ, однак для комфортної роботи та уникнення деградації продуктивності рекомендовано використовувати не менше 16 ГБ RAM, що дозволяє забезпечити паралельне функціонування всіх віртуальних машин без перевантаження ресурсів, при цьому дисковий простір має становити щонайменше 40 ГБ, а використання твердотільного накопичувача (SSD) суттєво підвищує швидкість доступу до даних та загальну продуктивність системи. Центральний процесор повинен підтримувати апаратну віртуалізацію (Intel VT-x або AMD-V) та мати не менше чотирьох фізичних ядер, що забезпечує можливість ефективного розподілу обчислювальних ресурсів між віртуальними середовищами та контейнеризованими процесами.

Розподіл ресурсів між окремими віртуальними машинами визначається їх функціональним призначенням і рівнем навантаження, при якому маршрутизатор pfSense потребує від 1 до 2 ГБ оперативної пам’яті, одного або двох віртуальних ядер процесора та приблизно 10 ГБ дискового простору, що є достатнім для виконання функцій маршрутизації та фільтрації трафіку, тоді як клієнтська система на базі Windows потребує вже більшого обсягу ресурсів — від 2 до 4 ГБ оперативної пам’яті, двох віртуальних ядер процесора та значно більшого дискового простору (приблизно 120 ГБ), що обумовлено особливостями самої операційної системи та необхідністю імітації реального користувацького середовища. Сервер на базі Ubuntu Server із Docker Engine займає проміжну позицію за ресурсними вимогами, використовуючи від 2 до 4 ГБ оперативної пам’яті, два віртуальних ядра процесора та близько 32 ГБ дискового простору, що дозволяє розгортати кілька контейнерів одночасно без втрати стабільності роботи,

причому самі контейнери споживають мінімальні ресурси завдяки легковаговій природі контейнеризації (див. рис. 1.9).

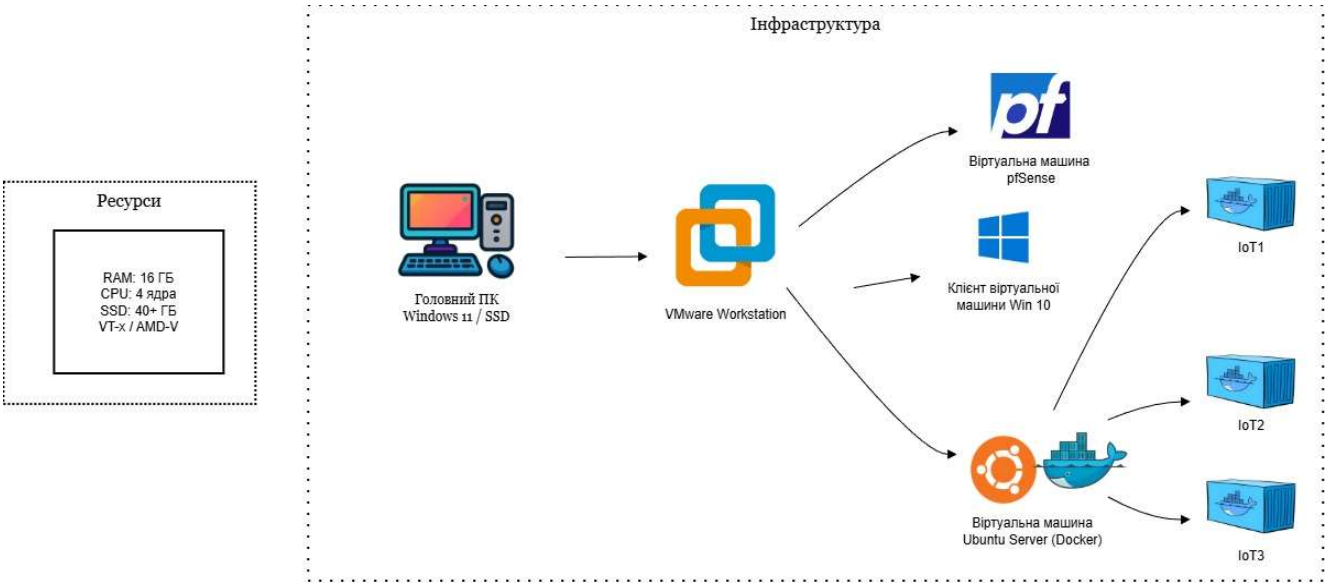


Рисунок 1.9 – Структура ресурсів віртуального стенду

З точки зору мережевих вимог важливою є підтримка віртуальних мереж типу VMnet, зокрема режимів NAT та Host-only, а також забезпечення коректної роботи Promiscuous mode для мережевих інтерфейсів, що є критично необхідним для функціонування драйвера "macvlan" у Docker, оскільки саме цей механізм дозволяє контейнерам отримувати власні MAC- та IP-адреси та поводитися як повноцінні вузли мережі, додатково необхідною є підтримка відповідних можливостей ядра Linux для роботи з macvlan, а також використання Docker Engine і Docker Compose для автоматизованого розгортання та управління контейнеризованими сервісами.

Обмеження такого середовища пов’язані передусім із залежністю від ресурсів хост-системи, що може призводити до зниження продуктивності при значному навантаженні або недостатньому обсязі оперативної пам’яті, а також із відсутністю апаратних міжмережевих екранів, що обмежує можливості моделювання деяких низькорівневих механізмів захисту.

2. ПРОЄКТУВАННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАХИЩЕНОГО СЕГМЕНТУ

2.1 Архітектурна модель запропонованого рішення

Після формування теоретичних засад забезпечення безпеки інфраструктури підключених пристроїв наступним етапом стає практична реалізація запропонованих підходів у вигляді експериментального стенду, основним призначенням якого є побудова такої архітектурної моделі, що одночасно відтворює умови функціонування реальної корпоративної мережі та забезпечує можливість тестування політик безпеки, гнучкого масштабування кількості вузлів і проведення досліджень без суттєвих витрат обчислювальних ресурсів, при цьому в основі рішення лежить поєднання технологій віртуалізації на базі VMware Workstation, програмного маршрутизатора pfSense та контейнеризації за допомогою Docker із використанням драйвера macvlan, що дозволяє створити повноцінну сегментовану мережу з чітким розмежуванням зон довіри та контрольованими каналами взаємодії між ними, де кожен елемент інфраструктури виконує визначену функцію, а вся система загалом функціонує як узгоджений механізм із централізованим управлінням трафіком (див. рис. 2.1) [20].

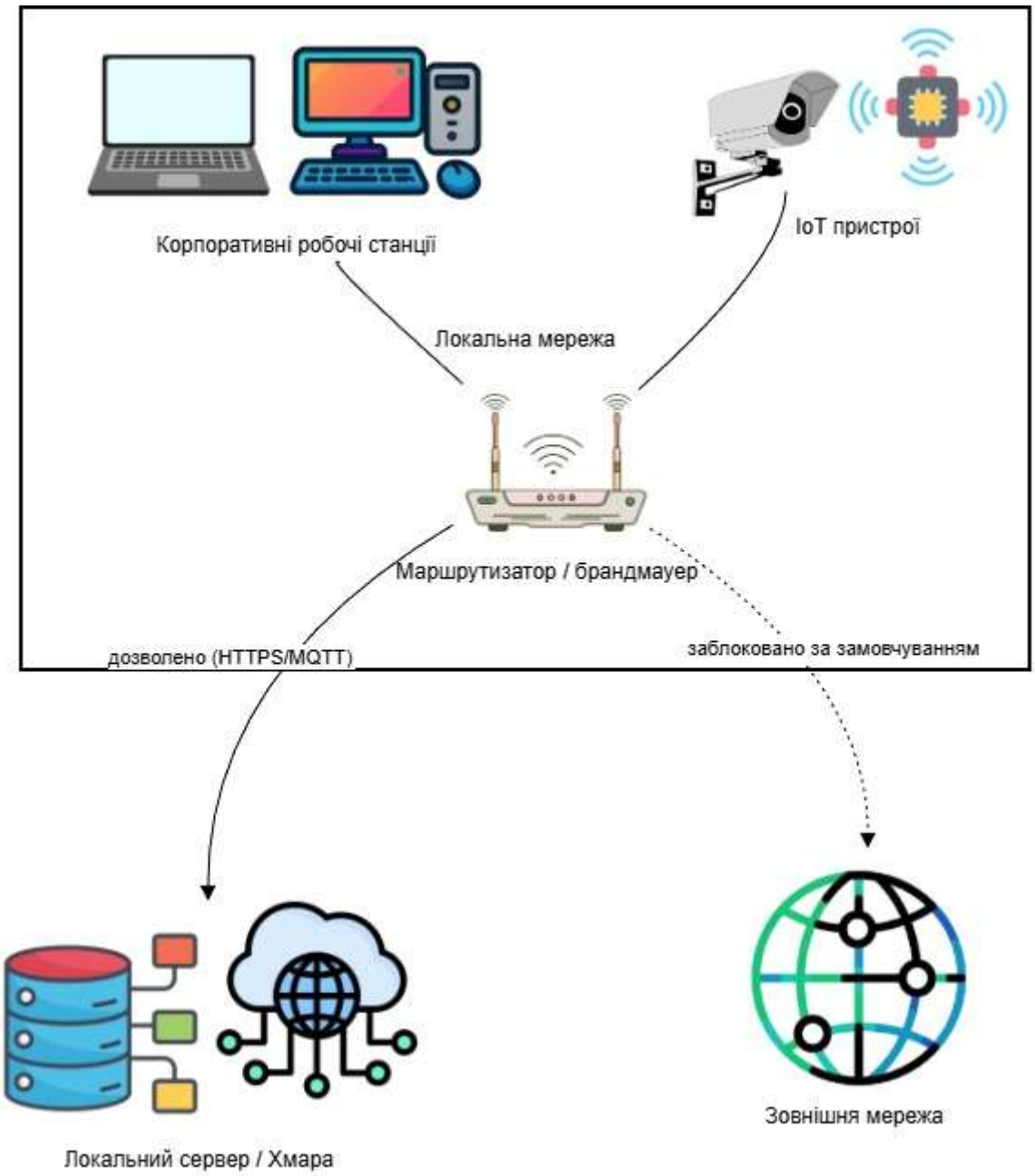


Рисунок 2.1 – Концептуальна архітектура захищеного IoT-сегменту

Структурна схема та логічні сегменти (VMnet). Розгляд структурної схеми стенду передбачає аналіз логічної сегментації середовища та політики IP-адресації, які формують основу для подальшої реалізації механізмів маршрутизації та фільтрації трафіку, причому побудована модель базується на трьох основних віртуальних вузлах, кожен із яких виконує власну роль у загальній архітектурі, серед яких перший вузол представлений у вигляді віртуального маршрутизатора на базі pfSense, що забезпечує функції маршрутизації, трансляції адрес та

міжмережевого екранування, другий вузол реалізовано у вигляді клієнтської машини на базі Windows 10, яка імітує корпоративний сегмент користувачів, тоді як третій вузол представлений сервером на базі Ubuntu Server із встановленим Docker Engine, який забезпечує імітацію IoT-пристроїв у вигляді контейнерів, що функціонують як окремі мережеві об’єкти з власними адресами та мережевими параметрами (див. рис. 2.2).

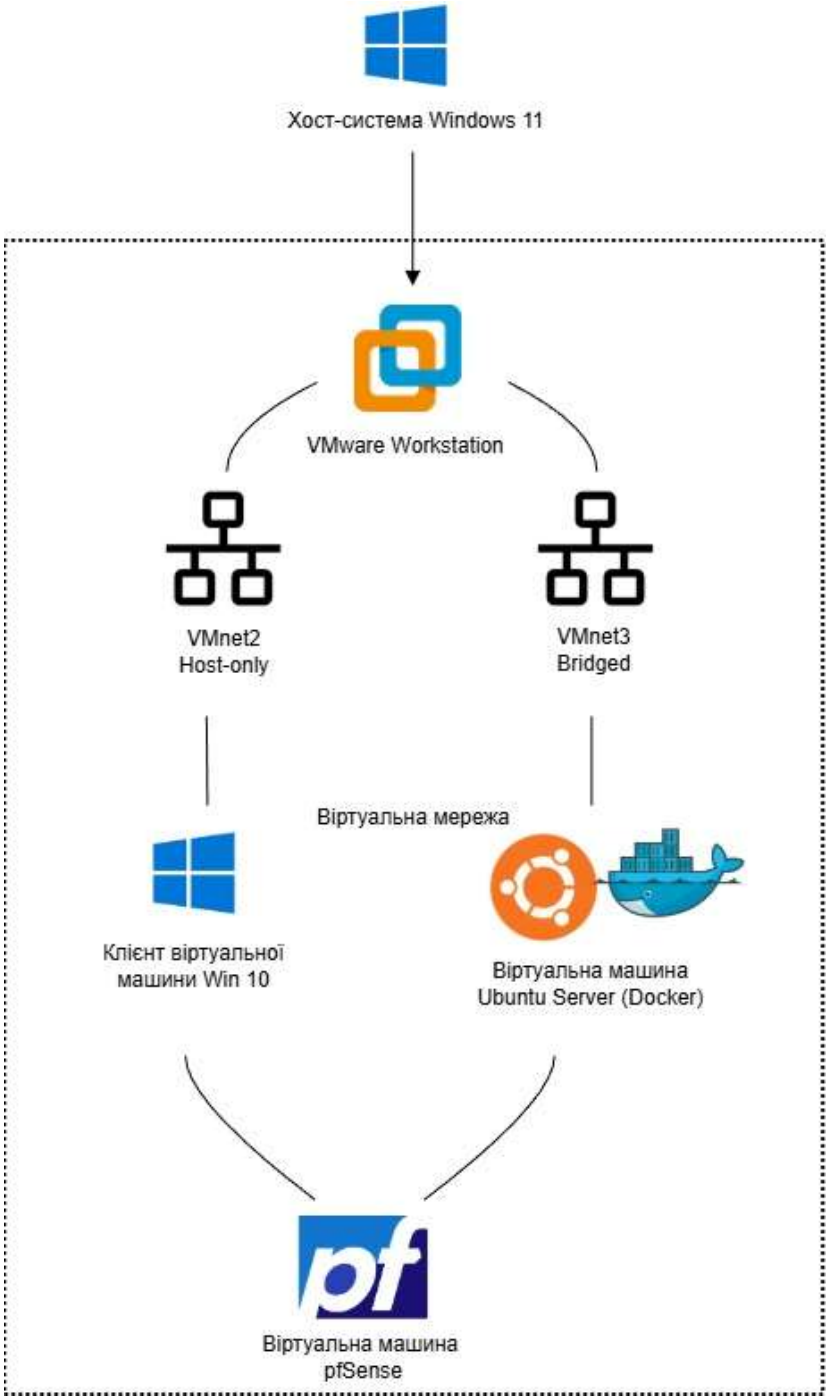


Рисунок 2.2 – Мережева топологія віртуального стенду

Об’єднання зазначених вузлів здійснюється у межах віртуального середовища за допомогою ізольованих мереж типу VMnet, які забезпечують логічну сегментацію, аналогічну використанню VLAN у фізичних мережах, при цьому в конфігурації VMware Workstation використовуються кілька типів віртуальних мереж, серед яких (див. рис. 2.3) [21]:

1. VMnet8 застосовується для організації доступу до глобальної мережі через механізм NAT;
2. VMnet2 виступає як ізольований сегмент корпоративної мережі для клієнтських пристроїв;
3. VMnet3 використовується як окремий сегмент для IoT-пристроїв.

У результаті чого кожна з мереж функціонує як незалежний широкомовний домен із власною адресною підмережею та обмеженим доступом до інших сегментів.

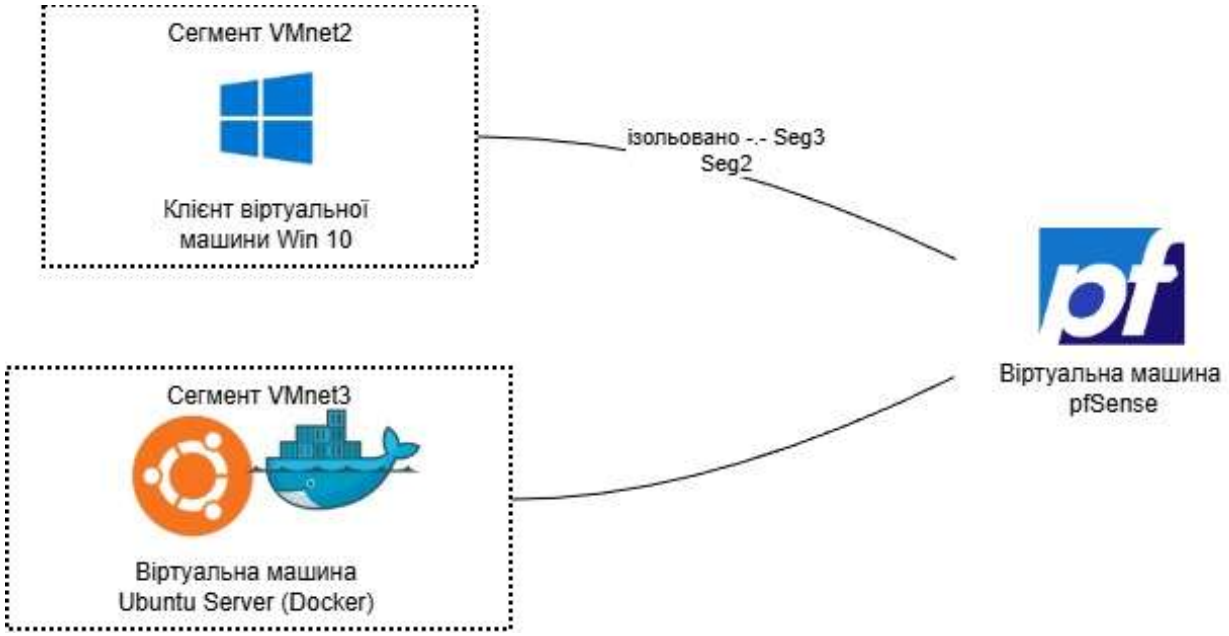


Рисунок 2.3 – Логічне розділення сегментів VMnet2 та VMnet3

Підключення віртуальних машин до відповідних мереж реалізується таким чином, що маршрутизатор pfSense має три мережеві інтерфейси, перший із яких підключений до VMnet8 та виконує роль WAN-інтерфейсу для доступу до Інтернету, другий підключений до VMnet2 та забезпечує взаємодію з корпоративним сегментом, а третій підключений до VMnet3 та обслуговує IoT-

сегмент, при цьому клієнтська машина на базі Windows 10 підключається до VMnet2, що дозволяє моделювати роботу користувачів у корпоративній мережі, тоді як сервер Ubuntu підключений до VMnet3, забезпечуючи ізольоване середовище для функціонування контейнеризованих IoT-вузлів, кількість яких може змінюватися залежно від потреб дослідження.

У межах IoT-сегменту контейнеризовані середовища функціонують як незалежні вузли мережі, кожен із яких отримує власну IP-адресу та взаємодіє з іншими контейнерами в межах одного широкомовного домену, що дозволяє імітувати поведінку реальних пристроїв без необхідності створення окремих віртуальних машин, у результаті чого забезпечується значна економія ресурсів та спрощується масштабування інфраструктури, при цьому взаємодія між сегментами повністю контролюється маршрутизатором pfSense, через який проходить увесь міжмережевий трафік, що дозволяє централізовано застосовувати політики безпеки та контролю доступу.

Запропонована архітектура реалізує багаторівневу модель ізоляції, у якій відсутній прямий канальний зв’язок між сегментами VMnet2 та VMnet3, що унеможливлює прямий обмін трафіком між корпоративними та IoT-пристроями, тоді як будь-яка взаємодія між ними можлива лише через маршрутизатор із відповідними правилами фільтрації, додатково ізоляція підтримується і на рівні самого IoT-сегменту, оскільки контейнери функціонують як окремі логічні вузли, що дозволяє моделювати розподілену інфраструктуру з великою кількістю пристроїв.

Переваги обраної архітектурної моделі полягають у її відповідності реальним умовам корпоративних мереж, можливості швидкого розгортання у межах однієї фізичної системи, гнучкості масштабування шляхом додавання нових контейнерів або віртуальних машин, а також у високому рівні керованості та контролю над мережевими потоками, що створює основу для подальшої реалізації механізмів маршрутизації, фільтрації та аналізу трафіку в межах захищеного сегменту.

Специфікація програмного маршрутизатора та адресація. У якості централізованого елементу мережевої інфраструктури обрано програмний

маршрутизатор на базі pfSense версії 2.8, що обумовлено його функціональними можливостями, серед яких [22]:

— підтримка stateful-фільтрації пакетів;
— наявність зручного веб-інтерфейсу для адміністрування;
— гнучка система списків контролю доступу;
— ведення журналів подій;
— детального аналізу мережевого трафіку.

Все це дозволяє реалізувати повноцінний контроль над усіма потоками даних у межах створеного середовища. Адресний простір у межах архітектурної моделі розподіляється між трьома основними сегментами відповідно до їх функціонального призначення, де зовнішній сегмент, що відповідає за доступ до глобальної мережі, використовує автоматичне призначення IP-адрес через DHCP у межах підмережі типу 192.168.x.x, що формується засобами гіпервізора, при цьому корпоративний сегмент організовано у вигляді підмережі 10.10.10.0/24 із шлюзом 10.10.10.1, від якого здійснюється адресація клієнтських вузлів, тоді як сегмент IoT-пристроїв виділено в окрему підмережу 10.20.20.0/24 із шлюзом 10.20.20.1, що забезпечує логічне відокремлення та контрольовану взаємодію між різними частинами інфраструктури.

У межах цієї схеми маршрутизатор pfSense виконує функції центрального вузла керування, маючи три мережеві інтерфейси, кожен із яких відповідає окремому сегменту, причому WAN-інтерфейс отримує IP-адресу автоматично через DHCP і забезпечує зв’язок із зовнішнім середовищем, LAN-інтерфейс має адресу 10.10.10.1 і обслуговує корпоративну мережу, тоді як інтерфейс OPT1 із адресою 10.20.20.1 відповідає за сегмент IoT, що дозволяє централізовано керувати маршрутизацією та політиками доступу між цими зонами. Клієнтська система на базі Windows 10 отримує адресу в межах корпоративної підмережі, наприклад 10.10.10.10, що дозволяє імітувати типовий користувацький вузол, тоді як сервер на базі Ubuntu Server використовує статичну адресу 10.20.20.5 у сегменті IoT, забезпечуючи платформу для запуску контейнеризованих сервісів, кожен із яких

отримує унікальну IP-адресу в діапазоні 10.20.20.11–10.20.20.15, що дозволяє розглядати їх як окремі мережеві пристрої (див. рис. 2.4).

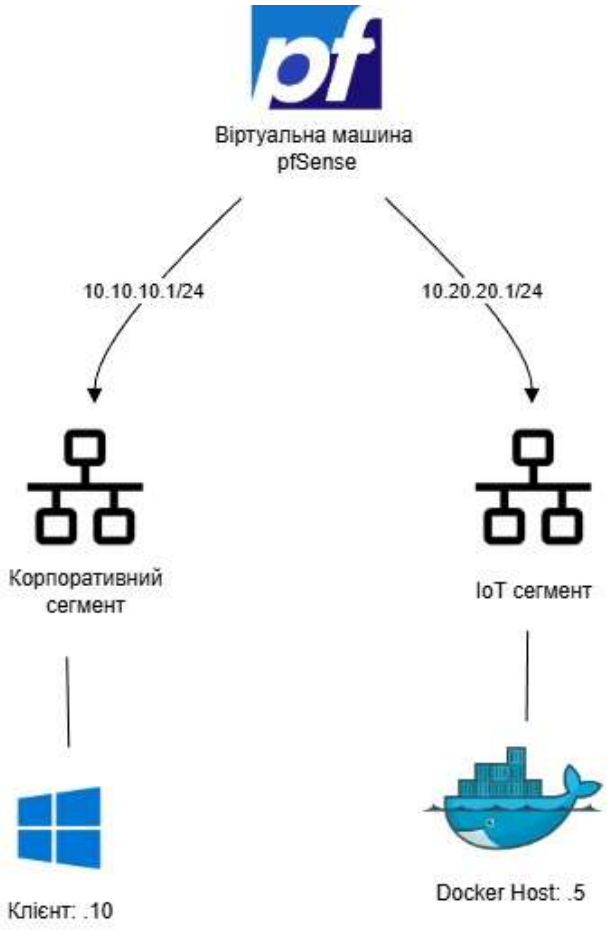


Рисунок 2.4 – Схема IP-адресації мережі

Така схема адресації забезпечує чітке розмежування мережевих зон і дозволяє однозначно визначати джерело та призначення трафіку під час обробки пакетів, при цьому логіка маршрутизації передбачає обробку кожного пакета відповідно до його походження та цільової адреси, де у випадку надходження трафіку з корпоративного сегменту виконується перевірка адреси призначення, і якщо вона належить до підмережі IoT, пакет відхиляється відповідно до політики ізоляції, тоді як у разі звернення до зовнішніх ресурсів пакет транслюється через механізм NAT і направляється у глобальну мережу, що забезпечує контрольований доступ до Інтернету. Для сегменту IoT застосовується політика повного обмеження вихідного трафіку, яка передбачає блокування будь-яких ініційованих з’єднань, що

унеможливлює несанкціонований доступ до корпоративної мережі або зовнішніх ресурсів, при цьому всі рішення щодо обробки пакетів приймаються централізовано на маршрутизаторі.

Особливістю реалізації IoT-сегменту є використання технології "macvlan" у середовищі Docker, що дозволяє контейнерам отримувати власні MAC- та IP-адреси в межах підмережі 10.20.20.0/24 і функціонувати як повноцінні мережеві вузли, які безпосередньо взаємодіють на канальному рівні в межах свого сегменту, що забезпечує високу точність моделювання реальної інфраструктури без необхідності створення окремих віртуальних машин для кожного пристрою. Обрана модель адресації характеризується простотою адміністрування, чіткою структурованістю та можливістю масштабування, оскільки додавання нових пристроїв або сегментів не потребує суттєвих змін у базовій конфігурації, при цьому відповідність реальним корпоративним практикам дозволяє використовувати розроблену архітектуру як основу для подальшого впровадження у фізичних мережах (див. табл. 2.1).

Таблиця 2.1 – Адресація та логічна сегментація віртуального середовища

Сегмент	VMnet	IP-мережа	Gateway	Основні вузли	Призначення
WAN	VMnet8	192.168.x.x	DHCP	pfSense WAN	Доступ до Internet через NAT
Corporate	VMnet2	10.10.10.0/24	10.10.10.1	Windows 10 (10.10.10.10)	Корпоративний сегмент
IoT	VMnet3	10.20.20.0/24	10.20.20.1	Ubuntu Server (10.20.20.5), Docker containers (10.20.20.11–15)	Ізольований сегмент IoT-пристроїв

2.2 Етапи побудови віртуального мережевого середовища

Після формування архітектурної моделі захищеної мережі наступним логічним етапом є її практична реалізація у віртуальному середовищі, що передбачає створення повноцінного лабораторного стенду, здатного відтворити сегментовану корпоративну інфраструктуру з можливістю подальшого тестування політик безпеки та перевірки працездатності окремих компонентів, при цьому процес розгортання виконується послідовно з поступовим введенням кожного елементу системи в експлуатацію, що дозволяє контролювати коректність налаштувань і уникати накопичення помилок перед переходом до більш складних конфігурацій.

Розгортання pfSense та клієнтської машини. Початковим етапом реалізації виступає розгортання центрального вузла інфраструктури у вигляді програмного маршрутизатора на базі pfSense у середовищі VMware Workstation, де створюється нова віртуальна машина з операційною системою на базі FreeBSD та базовими параметрами, достатніми для виконання функцій маршрутизації та фільтрації трафіку, при цьому ключовим моментом конфігурації є додавання трьох віртуальних мережевих адаптерів, кожен із яких відповідає окремому сегменту мережі, де (див. рис. 2.5):

1. перший адаптер підключається до VMnet8 і використовується як WAN-інтерфейс для доступу до зовнішньої мережі через NAT;
2. другий адаптер підключається до VMnet2 і забезпечує взаємодію з корпоративним сегментом;
3. третій адаптер підключається до VMnet3 і обслуговує ізольований сегмент IoT-пристроїв.

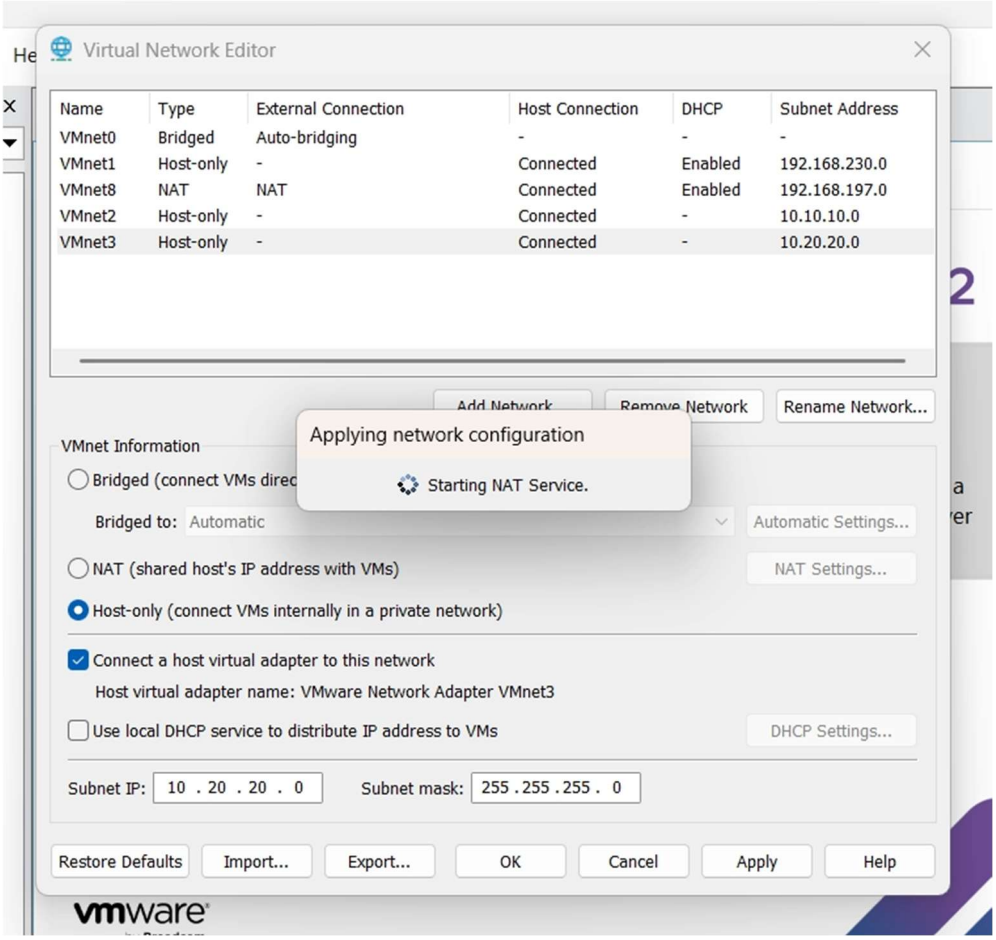


Рисунок 2.5 – Налаштування мережевих адаптерів у VMware

У результаті чого формується базова трирівнева модель сегментації, що відповідає попередньо визначеним принципам побудови мережі.

Після запуску віртуальної машини виконується первинне налаштування маршрутизатора через консольне меню, у межах якого здійснюється призначення мережевих інтерфейсів відповідно до їх функціонального призначення, де [23]:

1. інтерфейс em0 використовується як WAN;
2. em1 як LAN для корпоративної мережі;
3. em2 як OPT1 для IoT-сегменту.

Далі виконується конфігурація IP-адрес для кожного інтерфейсу, що передбачає автоматичне отримання адреси для WAN через DHCP, встановлення статичної адреси 10.10.10.1/24 для LAN та 10.20.20.1/24 для OPT1, що дозволяє сформувати основу для подальшої маршрутизації між сегментами та забезпечити коректну адресацію всіх вузлів у мережі (див. рис. 2.6).

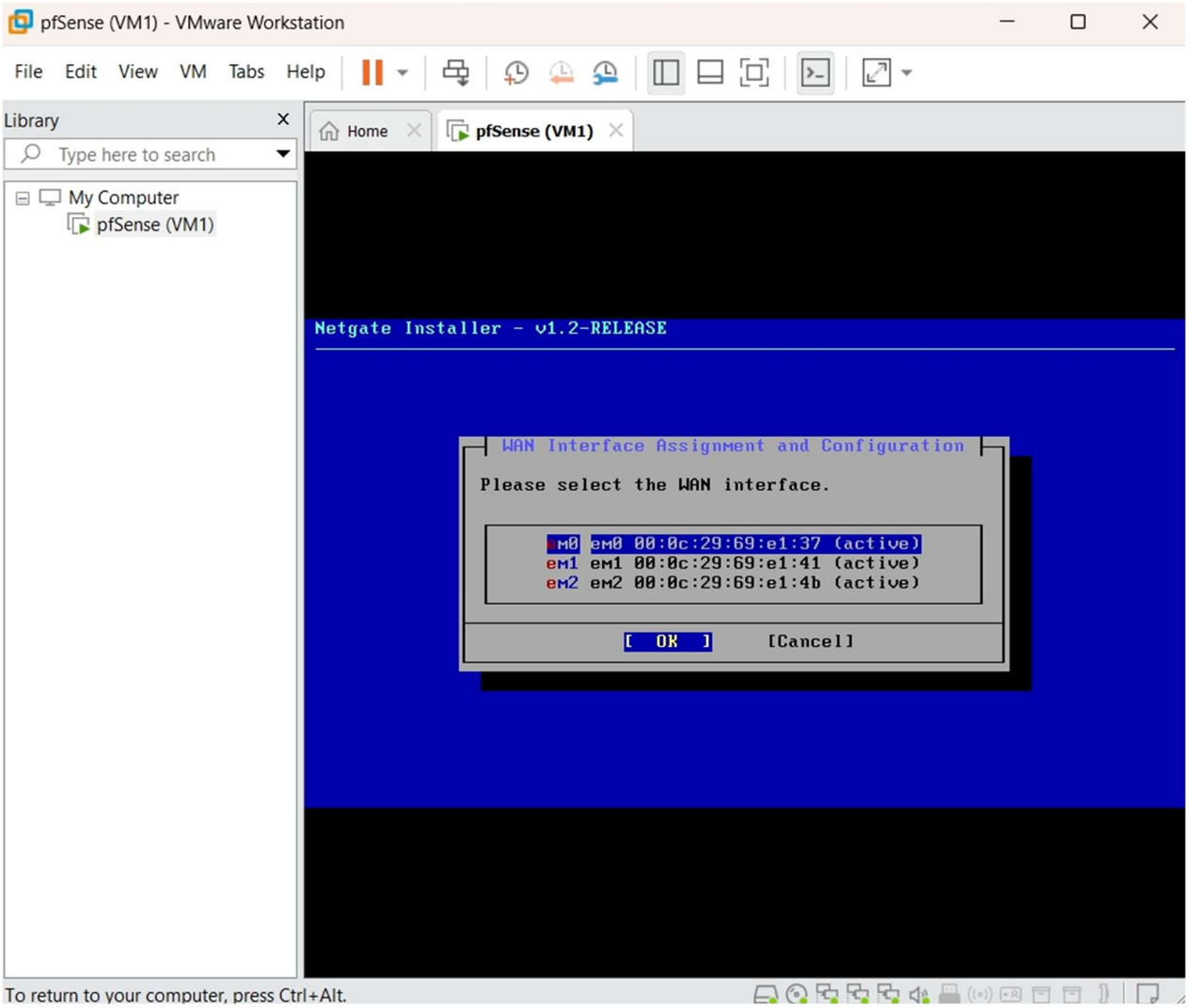


Рисунок 2.6 – Консольна конфігурація pfSense

Для прискорення процесу налаштування базових політик доступу використовується консольна утиліта "easyrule", яка дозволяє оперативно створювати правила міжмережевого екрану без необхідності використання веб-інтерфейсу, при цьому формуються дозволяючі правила для корпоративного сегменту, що забезпечують доступ до зовнішніх ресурсів через протоколи HTTP та HTTPS, а також дозволяють виконання ICMP-запитів для перевірки доступності мережевих вузлів, у результаті чого реалізується базова політика доступу, яка дозволяє вихідний трафік із корпоративної мережі, тоді як для інтерфейсу IoT за замовчуванням зберігається повне блокування будь-яких з’єднань, що автоматично реалізує принцип заборони всього трафіку, який явно не дозволений (див. рис. 2.7).

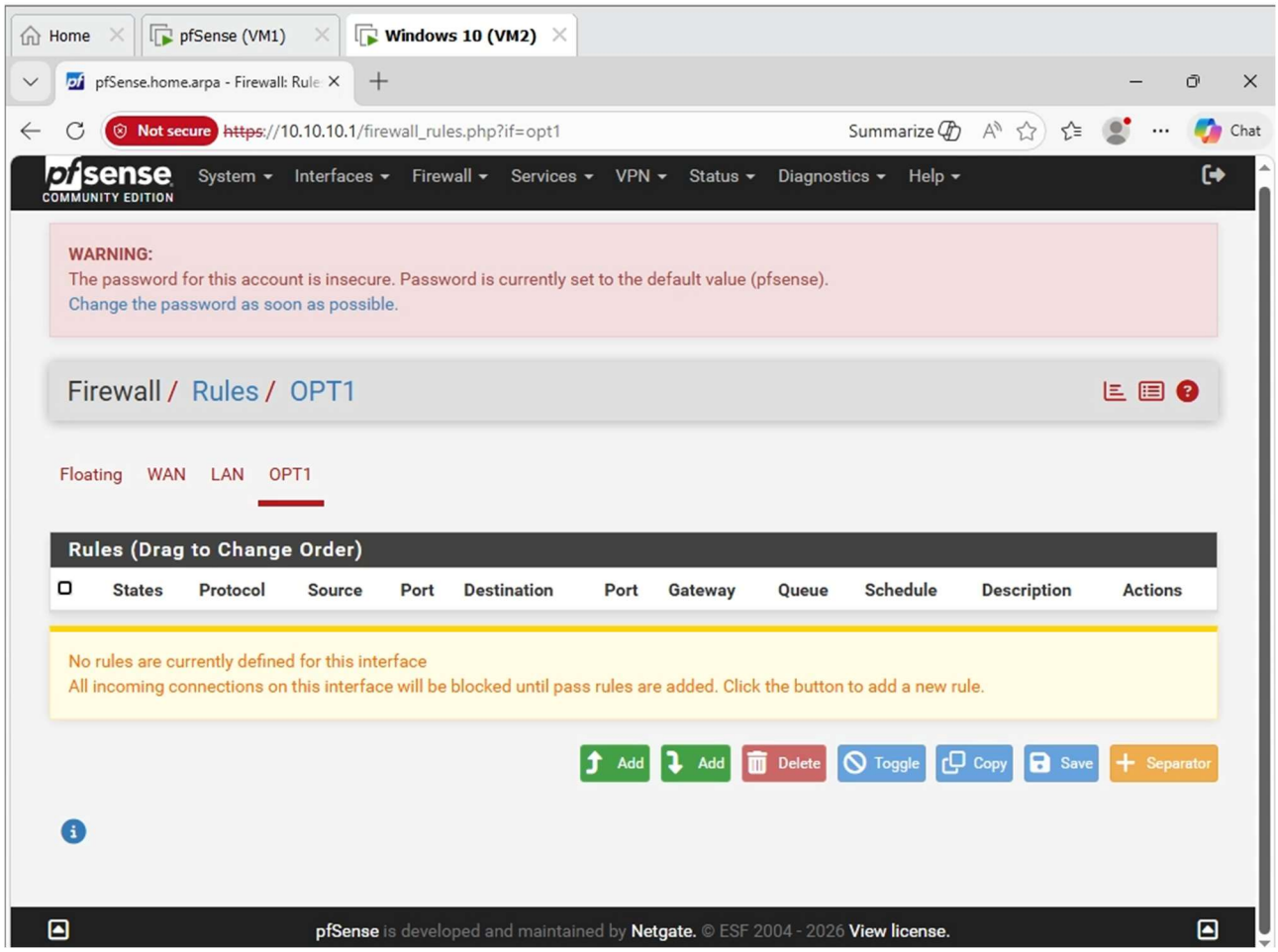


Рисунок 2.7 – Інтерфейс налаштування правил pfSense

Наступним етапом виконується розгортання клієнтської віртуальної машини на базі операційної системи Windows 10, яка використовується для імітації робочого місця користувача у корпоративному середовищі, при цьому для неї виділяються відповідні апаратні ресурси, зокрема оперативна пам’ять у розмірі близько 4 ГБ, а також налаштовується підключення до мережі VMnet2, що дозволяє отримати IP-адресу у межах підмережі 10.10.10.0/24 із шлюзом 10.10.10.1, у результаті чого формується повноцінний клієнтський вузол, здатний взаємодіяти з маршрутизатором та зовнішніми ресурсами.

Перевірка працездатності розгорнутого сегменту виконується шляхом проведення базового тестування мережевої взаємодії, що передбачає виконання ICMP-запитів до шлюзу корпоративної мережі для підтвердження доступності маршрутизатора, а також тестування доступу до зовнішніх ресурсів шляхом звернення до публічних серверів, що дозволяє підтвердити коректність роботи

механізму трансляції мережевих адрес та загальної маршрутизації, при цьому логіка обробки пакетів передбачає послідовну перевірку правил фільтрації, де для кожного пакета, отриманого з корпоративного сегменту, виконується аналіз відповідності правилам доступу, після чого у випадку дозволу він направляється через механізм NAT до зовнішньої мережі, а у разі невідповідності політикам безпеки відхиляється, що забезпечує контрольований і безпечний обмін даними в межах створеної інфраструктури [23].

Налаштування імітації сенсорів (Ubuntu та Docker). Подальший етап розгортання середовища передбачає створення окремого вузла для імітації IoT-пристроїв на базі серверної операційної системи Ubuntu Server із використанням технологій контейнеризації, що дозволяє реалізувати велику кількість логічно незалежних сенсорів без значного навантаження на апаратні ресурси, при цьому для цього використовується друга віртуальна машина з операційною системою Ubuntu Server 22.04 LTS, яка підключається до ізольованого сегменту мережі та отримує статичну IP-адресу 10.20.20.5, що дозволяє однозначно ідентифікувати її як центральний вузол IoT-сегменту. Особливістю конфігурації мережевого інтерфейсу є необхідність підтримки передачі кадрів із різними MAC-адресами, що досягається завдяки використанню режиму, сумісного з роботою драйвера macvlan, оскільки саме цей механізм забезпечує коректну інтеграцію контейнерів у мережеве середовище як окремих вузлів [24].

Після завершення базового розгортання операційної системи виконується встановлення Docker Engine та Docker Compose за допомогою стандартних пакетних менеджерів, що передбачає оновлення списку пакетів і встановлення відповідних компонентів, у результаті чого Docker запускається як системний сервіс і готовий до використання для створення контейнеризованих середовищ, причому загальна концепція застосування "macvlan" полягає у створенні віртуальних мережевих інтерфейсів, які дозволяють кожному контейнеру отримати власну MAC-адресу та IP-адресу і функціонувати як повноцінний учасник мережі на канальному рівні, що є критично важливим для моделювання IoT-пристроїв у реалістичних умовах [25].

Конфігурація середовища контейнерів реалізується за допомогою файлу "docker-compose.yml", у якому описується набір сервісів, що відповідають окремим імітованим сенсорам, кожен із яких використовує легковаговий образ, працює у фоновому режимі та підключається до мережі типу "macvlan" із жорстко заданою IP-адресою в межах підмережі 10.20.20.0/24, при цьому мережа визначається із зазначенням фізичного інтерфейсу хоста як базового для створення віртуальних інтерфейсів, що забезпечує пряме включення контейнерів у сегмент IoT. Запуск цієї конфігурації здійснюється однією командою у фоновому режимі, після чого всі визначені контейнери автоматично створюються та переходять у робочий стан, формуючи групу логічно незалежних пристроїв із власними мережевими параметрами (див. рис. 2.8) [24].

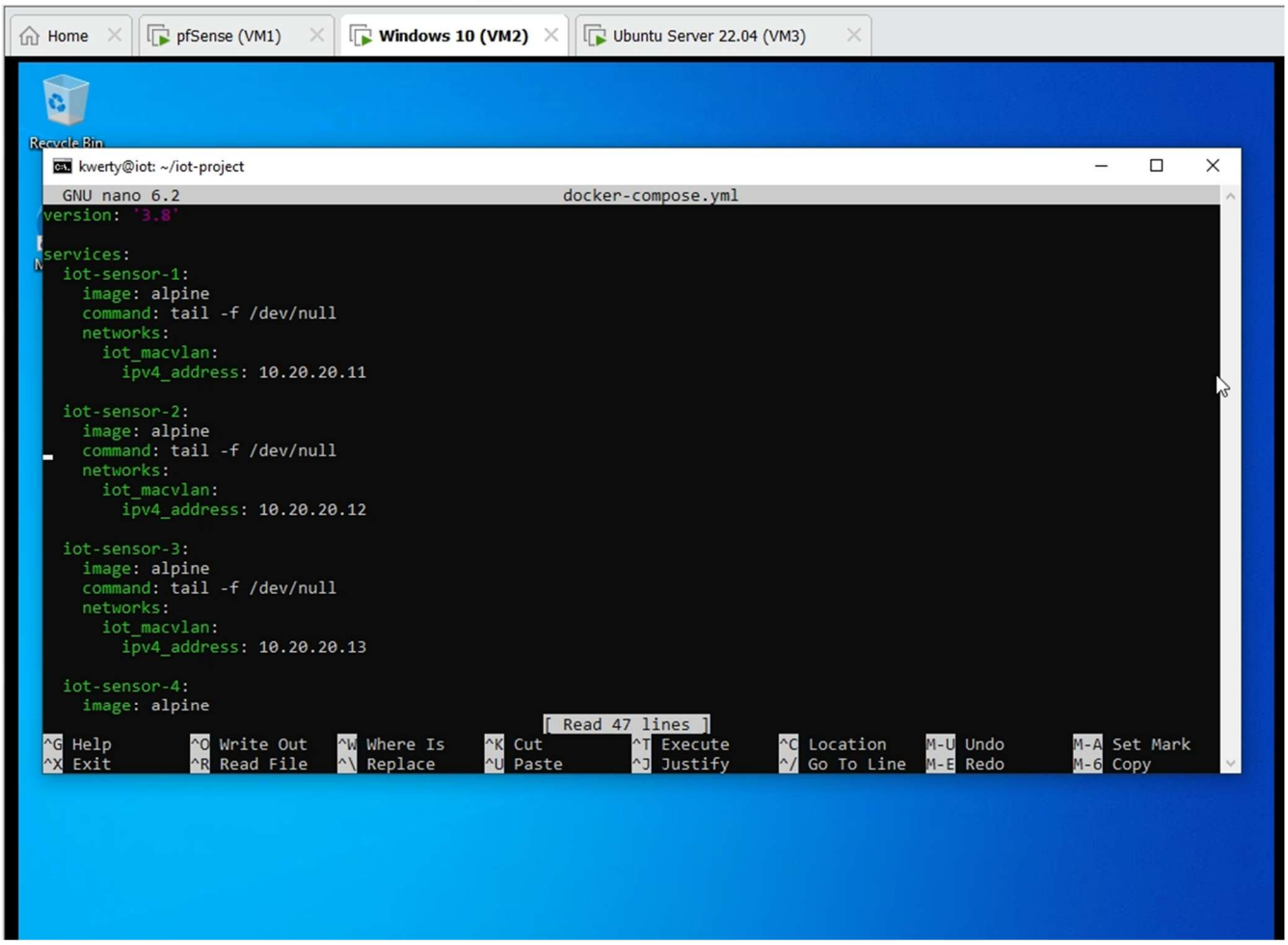


Рисунок 2.8 – Фрагмент docker-compose конфігурації

У результаті такої конфігурації формується чітка структура взаємодії, у межах якої сервер Ubuntu виступає базовою платформою з адресою 10.20.20.5, тоді як кожен контейнер отримує власну адресу, наприклад 10.20.20.11–10.20.20.15, що дозволяє здійснювати їх ідентифікацію та взаємодію між собою, при цьому перевірка коректності розгортання виконується за допомогою інструментів Docker, які дозволяють переглянути мережеві налаштування контейнерів і переконатися у правильності призначення IP-адрес та підключення до відповідної мережі. Логіка функціонування такої системи передбачає, що контейнери можуть взаємодіяти між собою на рівні L2 у межах одного сегменту, тоді як будь-який зовнішній трафік спрямовується до маршрутизатора, де підлягає обробці відповідно до встановлених політик безпеки (див. рис. 2.9).

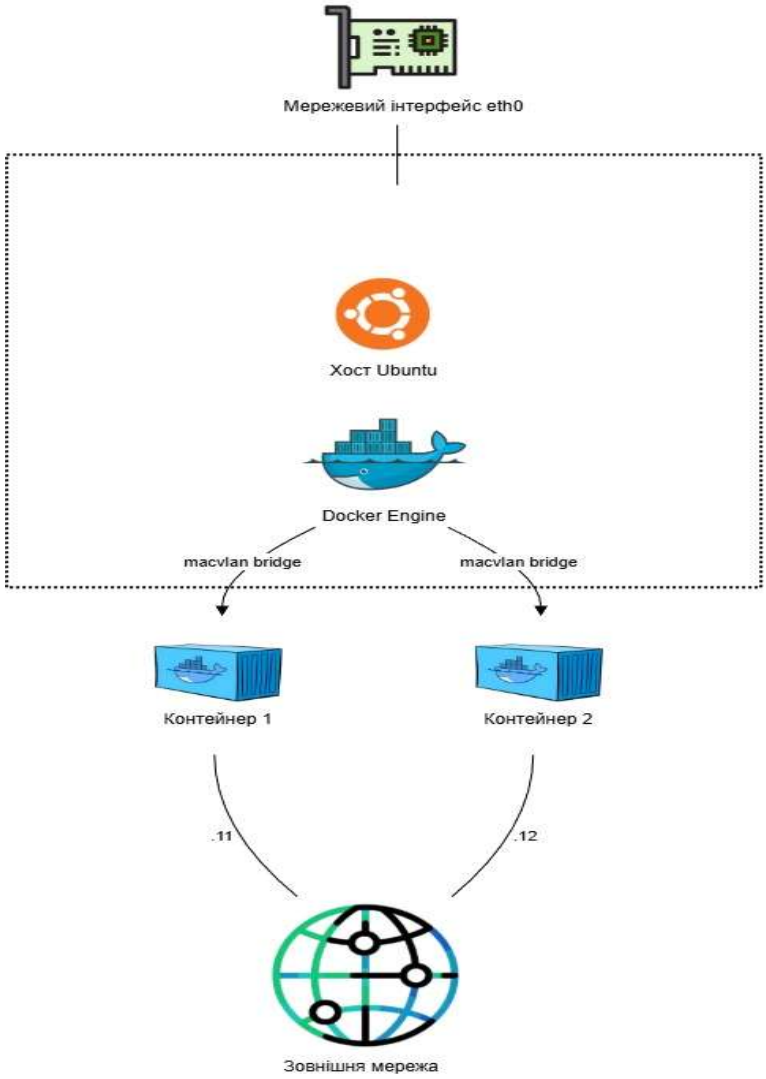


Рисунок 2.9 – Схема мережі "macvlan" у Docker

У випадку надходження пакета від одного з контейнерів до маршрутизатора виконується перевірка правил фільтрації, і за відсутності явного дозволу такий пакет відхиляється, що відповідає принципу повної ізоляції IoT-сегменту, тоді як дозволені запити можуть оброблятися відповідно до заданих правил, забезпечуючи контрольований обмін даними у межах інфраструктури (див. рис. 2.10).

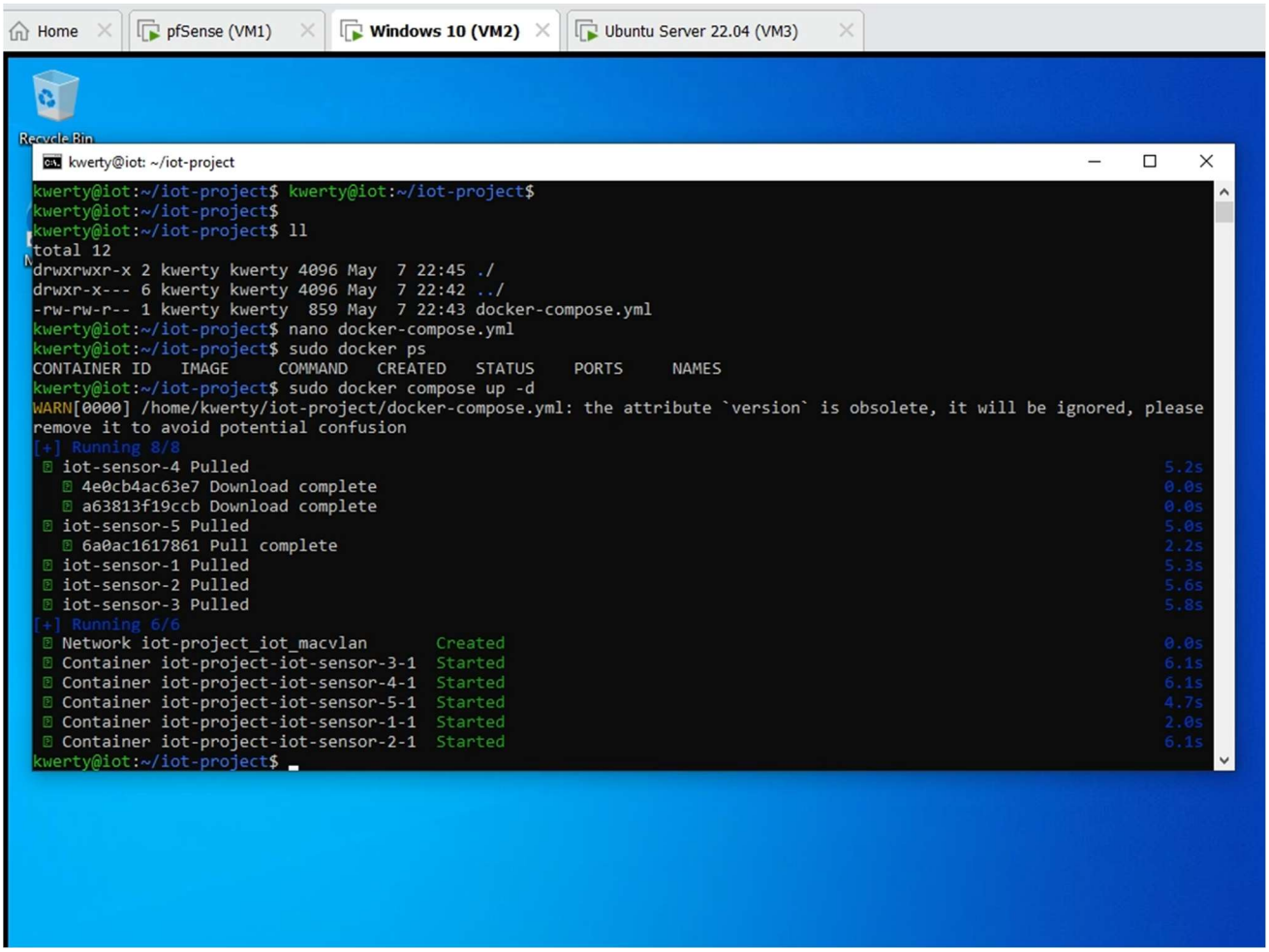


Рисунок 2.10 – Результат запуску контейнерів у терміналі

У результаті виконання всіх описаних етапів формується повноцінне віртуальне середовище, яке точно відтворює сегментовану корпоративну мережу з ізольованим IoT-сегментом, централізованим контролем доступу та можливістю гнучкого масштабування, причому використання Docker із драйвером macvlan дозволяє імітувати значну кількість пристроїв із мінімальними витратами ресурсів, що робить таку модель ефективною для досліджень і тестування механізмів

захисту, а отримане середовище стає основою для подальшого налаштування політик безпеки та перевірки ефективності ізоляції мережевих сегментів.

2.3 Алгоритми маршрутизації та конфігурація політик безпеки

Після завершення розгортання віртуального середовища визначальним етапом стає забезпечення його захищеності шляхом налаштування логіки обробки мережевих пакетів, у межах якої реалізується контроль доступу між сегментами та визначаються правила їх взаємодії, що дозволяє чітко розмежувати дозволені та заборонені канали обміну даними, при цьому в основі запропонованої архітектури централізованим елементом прийняття рішень виступає міжмережевий екран, реалізований на базі pfSense, який функціонує відповідно до принципів "stateful packet inspection", обробляє пакети з урахуванням їхнього стану та використовує механізм послідовної перевірки правил за принципом першого збігу, доповнений політикою повної заборони за замовчуванням, що дозволяє формалізувати обробку трафіку у вигляді чітко визначених алгоритмів, придатних для подальшого аналізу ефективності функціонування системи та візуалізації у вигляді блок-схем.

У межах реалізації політик безпеки виділяється два основних сценарії обробки трафіку, серед яких перший передбачає забезпечення контрольованого доступу корпоративного сегменту до зовнішніх ресурсів, тоді як другий спрямований на повну ізоляцію IoT-сегменту від інших мереж, що дозволяє мінімізувати ризики несанкціонованого доступу та поширення загроз у разі компрометації окремих пристроїв. Корпоративний сегмент із підмережею 10.10.10.0/24 розглядається як довірена зона, однак навіть у цьому випадку доступ до інших сегментів підлягає суворому контролю, де основним завданням є забезпечення доступу до Інтернету через механізм трансляції адрес, одночасно забороняючи будь-які спроби взаємодії з IoT-сегментом, що досягається шляхом формування відповідного набору правил фільтрації [26].

Алгоритм забезпечення доступу корпоративного сегменту. Логіка обробки пакетів у цьому сценарії передбачає послідовну перевірку кожного пакета, що надходить на інтерфейс корпоративної мережі, де спочатку виконується аналіз адреси джерела, і у випадку її належності до підмережі 10.10.10.0/24 обробка продовжується, інакше пакет відхиляється, далі виконується перевірка адреси призначення, і якщо вона належить до підмережі IoT 10.20.20.0/24, пакет блокується відповідно до політики ізоляції, тоді як у разі звернення до будь-яких інших адрес пакет пропускається та передається до зовнішньої мережі через механізм NAT, що забезпечує контрольований доступ до Інтернету. Реалізація цієї логіки на практиці передбачає створення правила блокування для трафіку з корпоративного сегменту до IoT-підмережі та правила дозволу для вихідного трафіку до будь-яких інших адрес, при цьому використання механізму автоматичного outbound NAT забезпечує трансляцію внутрішніх адрес у зовнішню адресу маршрутизатора, а зворотний трафік обробляється відповідно до таблиці станів, що дозволяє коректно встановлювати з’єднання із зовнішніми ресурсами (див. рис. 2.11).

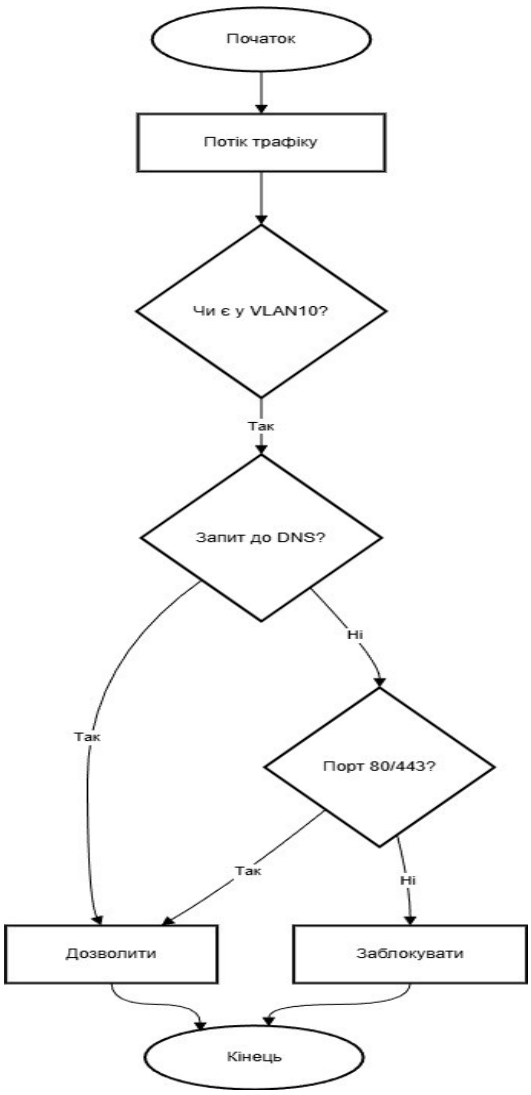


Рисунок 2.11 – Алгоритм обробки трафіку корпоративного сегменту

Особливість такого підходу полягає у мінімізації кількості необхідних правил при збереженні високого рівня контролю, оскільки обробка пакетів зупиняється на першому відповідному правилі, що знижує навантаження на систему та підвищує продуктивність маршрутизатора, водночас повна відсутність дозволів на взаємодію з IoT-сегментом виключає можливість несанкціонованого доступу до нього з корпоративної мережі. Додатково реалізована логіка забезпечує неможливість латерального переміщення між сегментами, оскільки будь-які спроби встановлення з’єднання між ними блокуються на рівні міжмережевого екрану, що підвищує загальний рівень безпеки інфраструктури.

У підсумку така конфігурація політик безпеки формує чітко визначений алгоритм маршрутизації, у межах якого кожен пакет проходить послідовну

перевірку відповідно до встановлених правил, що дозволяє досягти контрольованої взаємодії між сегментами, забезпечити доступ до зовнішніх ресурсів для довіреної зони та одночасно гарантувати повну ізоляцію IoT-пристроїв, що створює основу для подальшого тестування ефективності впроваджених механізмів захисту.

Алгоритм жорсткого блокування IoT-сегменту. Сегмент з адресним простором 10.20.20.0/24 розглядається як недовірене середовище, потенційно схильне до компрометації, у зв'язку з чим для нього застосовується максимально обмежувальна модель взаємодії, що передбачає повну ізоляцію як від інших внутрішніх сегментів, так і від зовнішніх мереж, із реалізацією політики "default deny", за якої будь-який мережевий пакет у разі відсутності явного дозволяючого правила автоматично відхиляється без формування відповіді, що унеможливлює отримання інформації про наявність або активність вузлів у цьому сегменті та водночас підвищує стійкість до сканування і мережевої розвідки.

Формалізація алгоритму обробки трафіку передбачає, що кожен пакет із джерелом у межах 10.20.20.0/24 при надходженні на інтерфейс перевіряється на наявність відповідного правила доступу, і за відсутності такого правила одразу відкидається без генерації відповіді, причому застосування саме silent drop дозволяє уникнути передачі ICMP-повідомлень, що додатково ускладнює визначення топології мережі потенційним зловмисником і мінімізує інформаційний витік на рівні мережевої взаємодії, а у випадку наявності дозволяючих правил, які в базовій моделі відсутні, пакет міг би бути оброблений відповідно до заданої політики, однак у запропонованій конфігурації така можливість свідомо не використовується з метою досягнення максимальної ізоляції.

Додатково передбачається блокування будь-яких спроб ініціювання з'єднань із IoT-сегменту до зовнішніх ресурсів, включаючи доступ до мережі Інтернет, що означає відсутність маршрутизації для вихідного трафіку та виключення можливості передачі даних за межі сегменту, а також унеможливлює використання пристроїв як частини ботнет-інфраструктури або каналів витоку інформації, причому взаємодія з іншими сегментами також повністю заборонена, що реалізує

логічну ізоляцію на рівні L3 і виключає будь-які горизонтальні переміщення між вузлами різних зон.

У разі необхідності допускається впровадження обмежених винятків, наприклад для забезпечення базових сервісів синхронізації часу або резолюції доменних імен через локальний шлюз, однак у базовій конфігурації такі винятки не застосовуються з метою збереження максимально жорсткої моделі безпеки, що дозволяє оцінити ефективність повної ізоляції без додаткових факторів впливу (див. рис. 2.12).

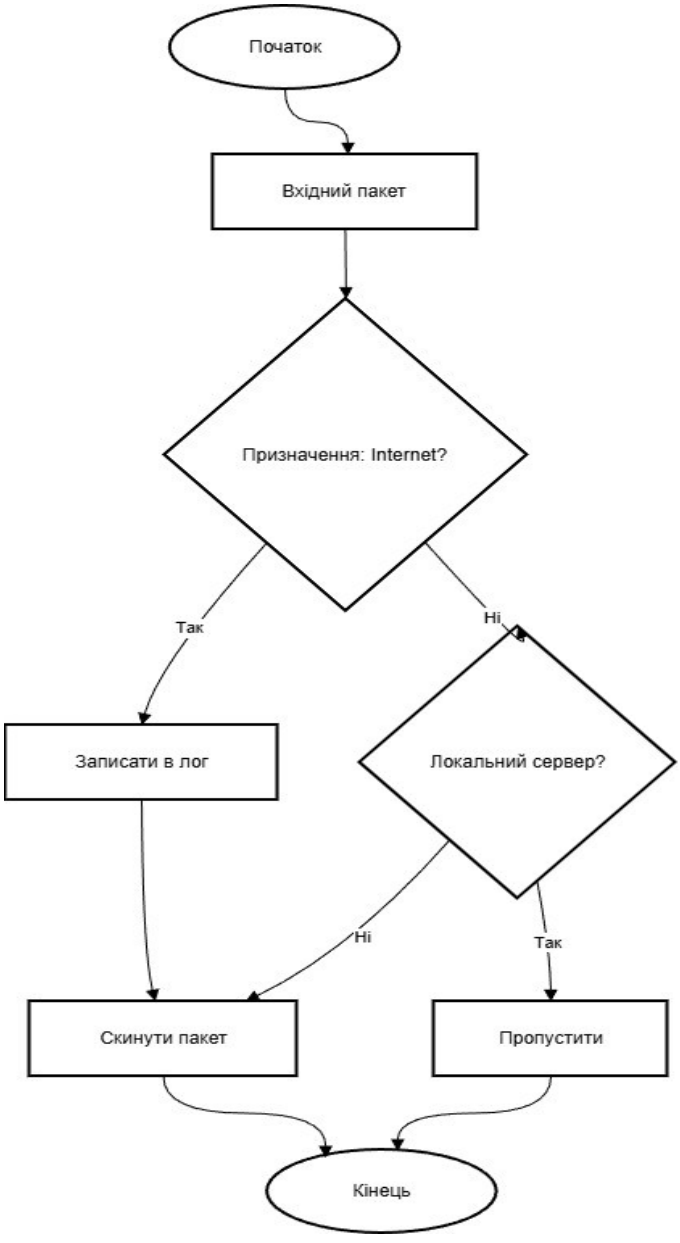


Рисунок 2.12 – Алгоритм блокування трафіку IoT

Така логіка організації мережевої взаємодії забезпечує приховування вузлів IoT-сегменту, унеможливлює їх виявлення стандартними методами сканування, мінімізує ризики несанкціонованого доступу та витоку даних, а також відповідає концепції Zero Trust, за якої жоден елемент інфраструктури не вважається довіреним за замовчуванням, і всі взаємодії підлягають строгому контролю або повному блокуванню, у результаті формується чітко визначена модель обробки мережевого трафіку, що поєднує контрольований доступ для довірених сегментів із повною ізоляцією недовірених зон і створює основу для подальшого аналізу ефективності реалізованих політик безпеки та перевірки стійкості мережевої інфраструктури до потенційних загроз.

2.4 Практична реалізація та тестування функціональності системи

Завершальний етап побудови захищеного сегменту корпоративної мережі полягає у практичній перевірці працездатності реалізованої архітектури та відповідності її функціонування вимогам безпеки, сформованим на етапі проектування, при цьому увага зосереджується не лише на коректності запуску окремих компонентів, але й на підтвердженні очікуваної поведінки міжмережевого екрану в різних сценаріях взаємодії між сегментами, що досягається шляхом проведення тестування у контрольованому віртуальному середовищі, яке дозволяє відтворити типові мережеві ситуації без ризику для реальної інфраструктури, а також застосуванням таких підходів, як аналіз таблиць маршрутизації, інспекція контейнерів і дослідження журналів подій міжмережевого екрану [27] .

Демонстрація працездатності та коректності адресації. Початковий етап тестування передбачає перевірку базової функціональності мережі, що включає доступність ресурсів та загальну коректність роботи механізмів віртуалізації, у результаті розгортання середовища за допомогою Docker Compose виконується аналіз створеної мережевої інфраструктури на сервері Ubuntu, де формується

структурований вивід із параметрами підмережі 10.20.20.0/24, визначеним шлюзом 10.20.20.1, прив’язкою до фізичного інтерфейсу ens33 та фіксацією IP-адрес контейнерів у діапазоні 10.20.20.11–10.20.20.15, що використовують драйвер "macvlan", одночасно здійснюється перевірка внутрішньої взаємодії між контейнерами, які функціонують в межах одного L2-домену, де забезпечується прямий обмін пакетами між вузлами, наприклад контейнер з адресою 10.20.20.11 має можливість взаємодії з іншими контейнерами, адреси яких послідовно зростають, що підтверджує коректність комутації в межах macvlan-мережі та демонструє цілісність і функціональність IoT-сегменту як ізольованого, але внутрішньо узгодженого середовища (див. рис. 2.13).

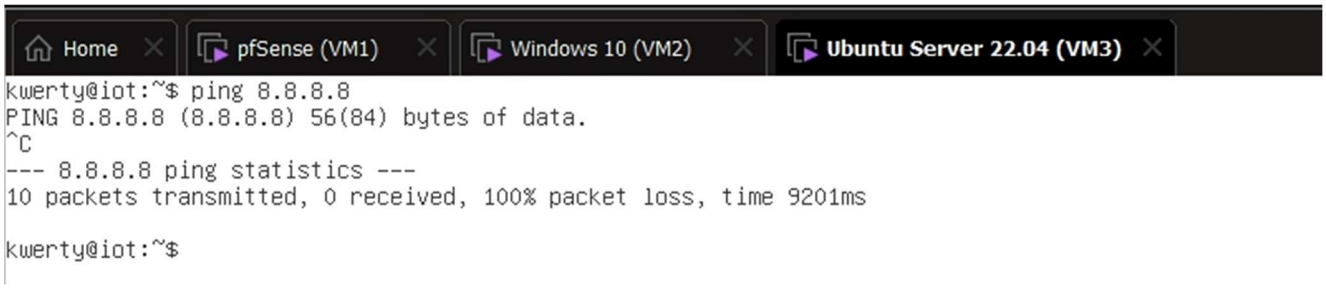


Рисунок 2.13 – Результат ping-запиту до Інтернету

Подальша перевірка спрямована на оцінку працездатності корпоративного сегменту, де клієнтська машина з операційною системою Windows 10, яка отримує IP-адресу 10.10.10.10 через DHCP, виконує тестування доступу до зовнішніх ресурсів шляхом надсилання ICMP-запитів до вузла 8.8.8.8, отримання відповідей підтверджує коректність маршрутизації через pfSense, правильну роботу механізму трансляції адрес (NAT) та наявність доступу до глобальної мережі, у такій схемі джерелом виступає клієнт корпоративного сегменту, проміжною ланкою є маршрутизатор, а кінцевим призначенням — зовнішній мережевий ресурс, що свідчить про відповідність функціонування сегменту заданій політиці доступу, де дозволяється вихідний трафік до Інтернету при одночасному обмеженні доступу до ізольованого IoT-сегменту (див. рис. 2.14).



Рисунок 2.14 – Вивід docker network inspect

Підтвердження ізоляції (ICMP-тестування). Наступний етап перевірки зосереджується на ключовому критерії ефективності всієї системи, яким виступає забезпечення повної мережевої ізоляції між сегментами, у зв’язку з чим тестування спрямовується на підтвердження відсутності небажаної взаємодії між корпоративною мережею та IoT-середовищем, передусім виконується спроба встановлення з’єднання з корпоративного сегменту до IoT-пристроїв, де логіка обробки пакетів відповідає заздалегідь визначеному алгоритму фільтрації, згідно з яким трафік, що надходить з підмережі 10.10.10.0/24, перевіряється на відповідність адресі призначення, і у випадку, коли ця адреса належить діапазону 10.20.20.0/24, пакети відхиляються міжмережевим екраном без подальшої обробки, тоді як у всіх інших випадках дозволяється їх передача до зовнішніх ресурсів, така поведінка підтверджує коректну реалізацію політики обмеження доступу до IoT-сегменту з боку корпоративної мережі (див. рис. 2.15).

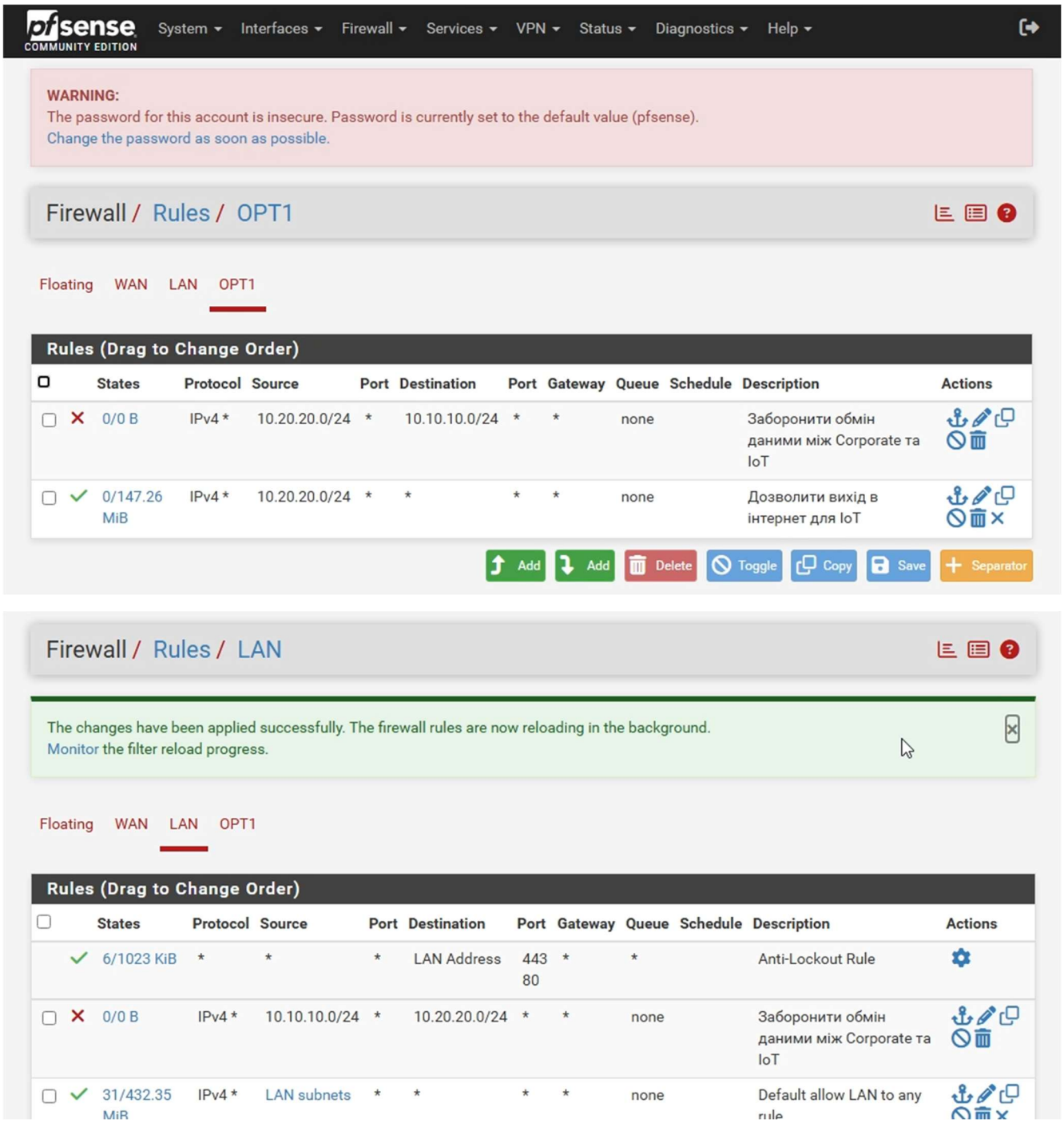


Рисунок 2.15 – Результат блокування між сегментами

Подальша перевірка здійснюється у зворотному напрямку шляхом ініціювання з’єднання з IoT-сегменту до корпоративної мережі, для цього виконується підключення до одного з контейнерів за допомогою команди доступу до його інтерактивної консолі, після чого здійснюється надсилання ICMP-запитів до адреси клієнтської машини у підмережі 10.10.10.0/24, відсутність відповіді свідчить про те, що IoT-пристрої не мають можливості ініціювати з’єднання з

корпоративним сегментом, що відповідає встановленим політикам безпеки та принципу повної ізоляції (див. рис. 2.16).

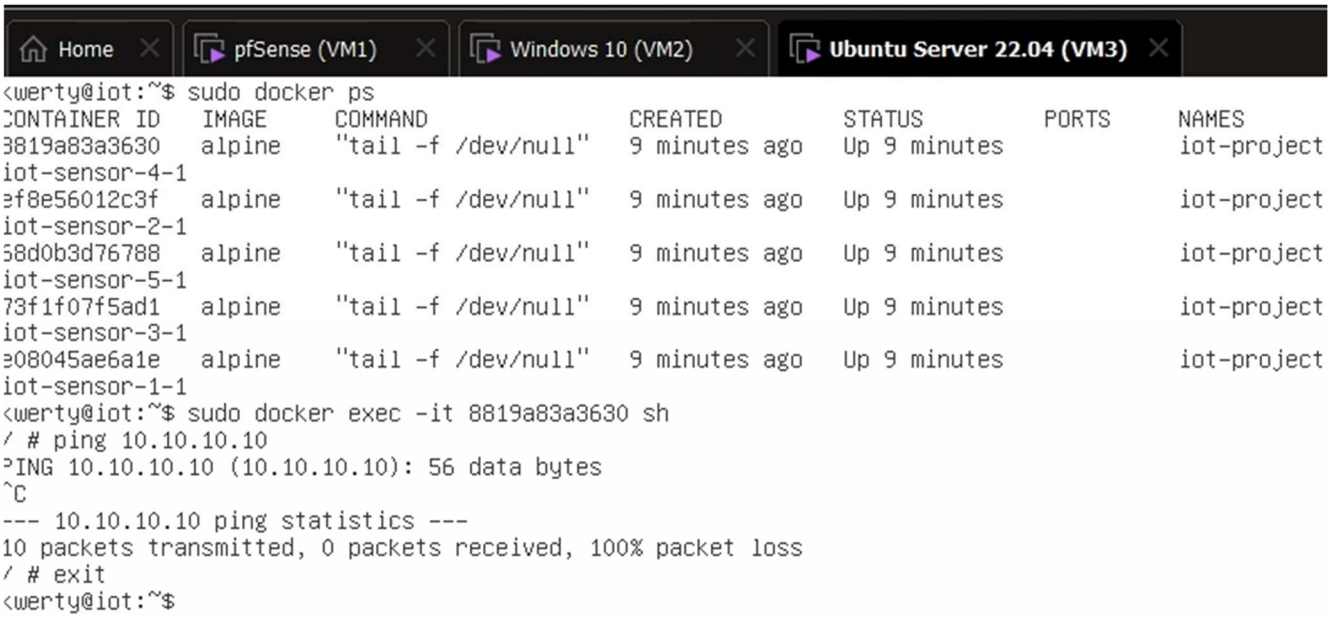


Рисунок 2.16 – Спроба доступу IoT до корпоративної мережі

Додатково виконується перевірка можливості доступу IoT-сегменту до глобальної мережі шляхом надсилання запитів до зовнішнього ресурсу, зокрема до вузла 8.8.8.8, у результаті чого також фіксується відсутність відповіді, що підтверджує повне блокування вихідного трафіку з IoT-пристроїв та їх ізоляцію не лише від корпоративної мережі, але й від Інтернету, таким чином досягається замкнутість сегменту, яка унеможливлює використання скомпрометованих пристроїв як точки доступу до зовнішніх ресурсів або каналів витоку даних.

Окрему увагу приділено аналізу журналів міжмережевого екрану, що дозволяє підтвердити факт блокування трафіку на рівні маршрутизатора, для цього використовується веб-інтерфейс системи pfSense, у якому в розділі журналів фіксуються записи про відхилені пакети із зазначенням інтерфейсу, джерела, призначення, протоколу та виконаної дії, у таких записах відображається, що пакети, які надходять з адрес корпоративного або IoT-сегменту, досягають маршрутизатора, обробляються відповідно до списків контролю доступу та відхиляються згідно з політикою безпеки, при цьому кожна така подія реєструється

у системі логування, що забезпечує можливість подальшого аудиту та аналізу мережевої активності (див. рис. 2.17) [27].

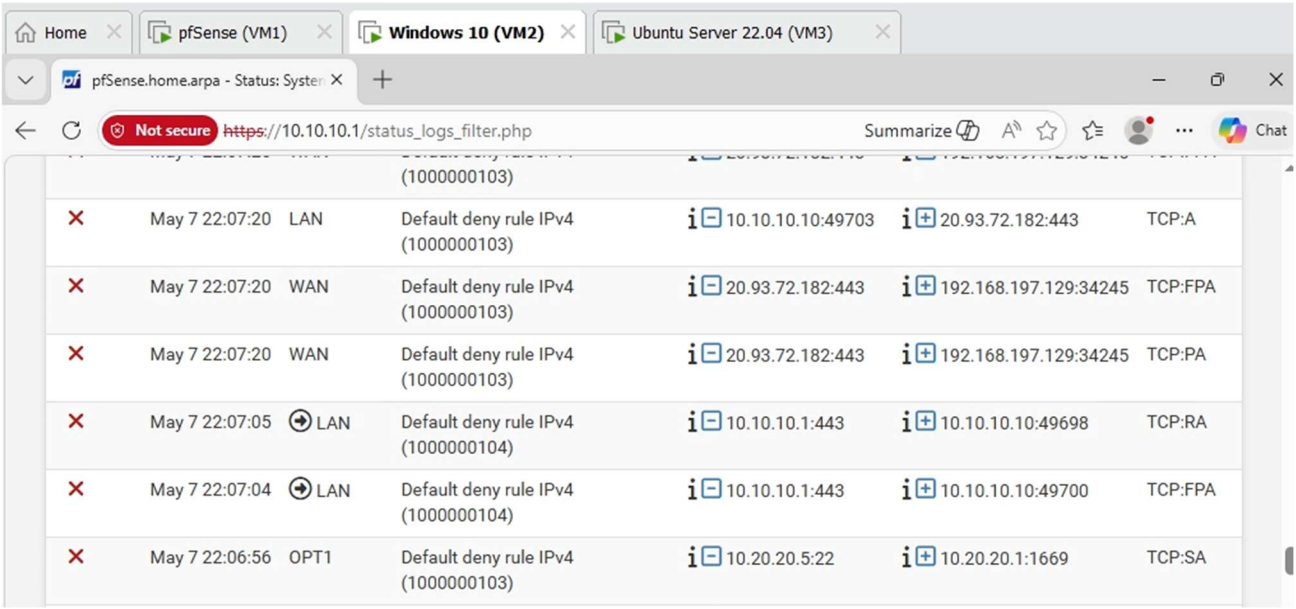


Рисунок 2.17 – Логи pfSense (Firewall Logs)

Отримані результати тестування дозволяють зробити узагальнений висновок про коректність функціонування всієї системи, адресація у всіх сегментах налаштована відповідно до визначеної схеми, контейнеризоване середовище успішно імітує незалежні IoT-пристрої, корпоративний сегмент має контрольований доступ до глобальної мережі, тоді як IoT-сегмент залишається повністю ізольованим як від внутрішніх ресурсів підприємства, так і від зовнішніх мереж, міжмережевий екран функціонує відповідно до принципу заборони за замовчуванням, а всі спроби порушення встановлених політик фіксуються у журналі, що підтверджує відповідність реалізованої архітектури вимогам безпеки, сформульованим на попередніх етапах, і дозволяє розглядати її як валідну модель захищеної корпоративної інфраструктури з інтегрованими IoT-пристроями.

3. ОПТИМІЗАЦІЯ, ПОРІВНЯННЯ ТА ПЕРСПЕКТИВИ РОЗВИТКУ

3.1 Методи оптимізації роботи мережевого екрану та усунення недоліків

Після реалізації базової архітектури захищеного IoT-сегменту та підтвердження її працездатності виникає логічна необхідність переходу до етапу оптимізації, оскільки навіть коректно побудована система мережевої ізоляції у віртуальному середовищі з часом може демонструвати обмеження, пов’язані з продуктивністю, масштабованістю та ефективністю обробки трафіку, що обумовлено специфікою використання ресурсів у середовищі віртуалізації, де одночасно функціонують гіпервізор, віртуальні машини та контейнеризовані сервіси, у зв’язку з цим виникає потреба перегляду підходів до організації обробки мережевих пакетів і розподілу системних ресурсів з метою забезпечення стабільної роботи при зростанні інтенсивності мережевих запитів та кількості імітованих IoT-пристроїв [28].

Програмна корекція та оптимізація правил доступу. Основою підвищення ефективності міжмережевого екрану виступає правильна організація правил фільтрації трафіку, де використовується модель послідовної перевірки умов зверху вниз із зупинкою на першому збігу, що відповідає принципу First Match, у такій моделі кожен пакет проходить перевірку на відповідність правилам доступу, і після знаходження першого релевантного запису подальший аналіз припиняється з негайним застосуванням відповідної дії, у випадку неефективного розташування правил система змушена виконувати зайві перевірки, що призводить до збільшення навантаження на процесор маршрутизатора та підвищення затримок обробки трафіку, оптимізація досягається шляхом зміни порядку правил таким чином, щоб найбільш часті сценарії оброблялися першими, наприклад на початку списку розміщуються правила блокування широкомовного та мультикаст-трафіку з IoT-сегменту, далі розташовуються дозволи для найбільш поширених типів

корпоративного трафіку, таких як HTTPS (TCP 443) та DNS (UDP/TCP 53), після чого додаються правила для діагностичних протоколів, включаючи ICMP, а завершальним елементом виступає правило заборони за замовчуванням, що відхиляє будь-які пакети, які не підпадають під жодну з попередніх умов, у такій логіці кожен вхідний пакет проходить послідовну перевірку, де у разі відповідності першому правилу обробка завершується негайно, а за відсутності збігу здійснюється перехід до наступного правила, що дозволяє значно скоротити середню кількість перевірок для кожного пакета, знизити навантаження на центральний процесор, мінімізувати затримки та підвищити загальну стабільність системи при збільшенні кількості IoT-пристроїв, особливо в умовах високої інтенсивності коротких телеметричних запитів (див. рис. 3.1) [28].

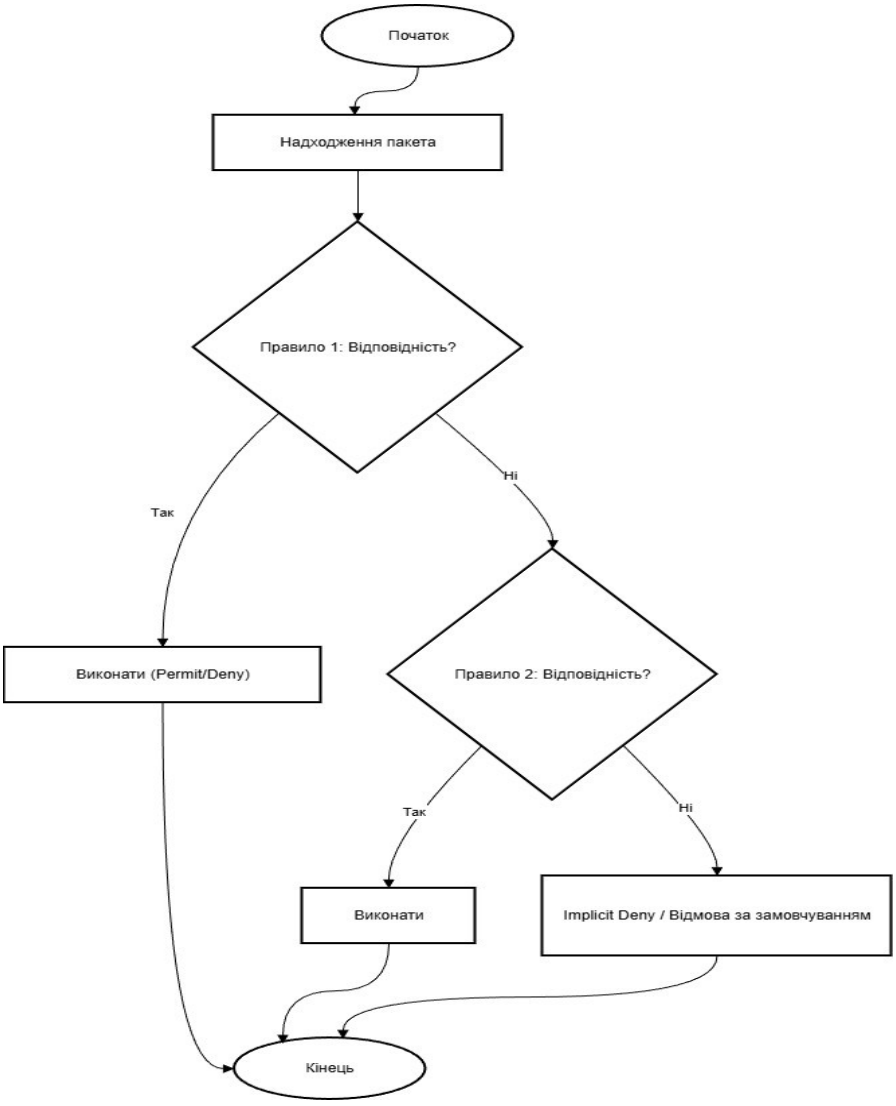


Рисунок 3.1 – Порядок обробки правил ACL

Оптимізація ресурсів імітаційного середовища. Важливим напрямом оптимізації виступає ефективне використання ресурсів хост-системи та віртуального середовища, оскільки розгортання стенду здійснюється на базі гіпервізора з одночасним використанням декількох віртуальних машин і контейнерної платформи, що формує сумарне навантаження на оперативну пам’ять і процесор, яке повинно розподілятися рівномірно між усіма компонентами інфраструктури, включаючи:

— маршрутизатор;

— клієнтську систему;

— сервер контейнерів із імітованими IoT-пристроями.

За відсутності контрольованого підходу до споживання ресурсів виникає ризик перевантаження оперативної пам’яті або процесора, що особливо проявляється при збільшенні кількості контейнерів у межах IoT-сегменту, незважаючи на те, що контейнеризація є значно легшою альтернативою повній віртуалізації, значна кількість одночасно запущених контейнерів здатна створювати суттєве навантаження, а неконтрольовані процеси можуть призводити до вичерпання пам’яті та виникнення станів типу "Out of Memory", що в свою чергу негативно впливає на стабільність роботи віртуальної машини та всієї системи загалом.

Ефективним способом усунення таких недоліків виступає впровадження обмежень на використання ресурсів на рівні контейнерів, що реалізується шляхом внесення відповідних параметрів у конфігураційний файл середовища, де для кожного імітованого IoT-пристрою задається фіксований ліміт оперативної пам’яті, у межах якого дозволяється функціонування контейнера, подібний підхід запобігає неконтрольованому зростанню використання ресурсів і забезпечує стабільність роботи системи навіть у випадку програмних збоїв окремих компонентів, при цьому загальне навантаження визначається пропорційно кількості контейнерів, де кожен із них споживає обмежений обсяг пам’яті, наприклад у діапазоні 32–50 МБ, що дозволяє прогнозовано оцінити сумарне використання ресурсів і забезпечити контрольоване масштабування без перевищення допустимих меж, у такій моделі кожен контейнер функціонує як

окремий вузол із жорстко визначеними ресурсними рамками, що виключає ситуації неконтрольованого витоку пам’яті та перевантаження системи (див. рис. 3.2).

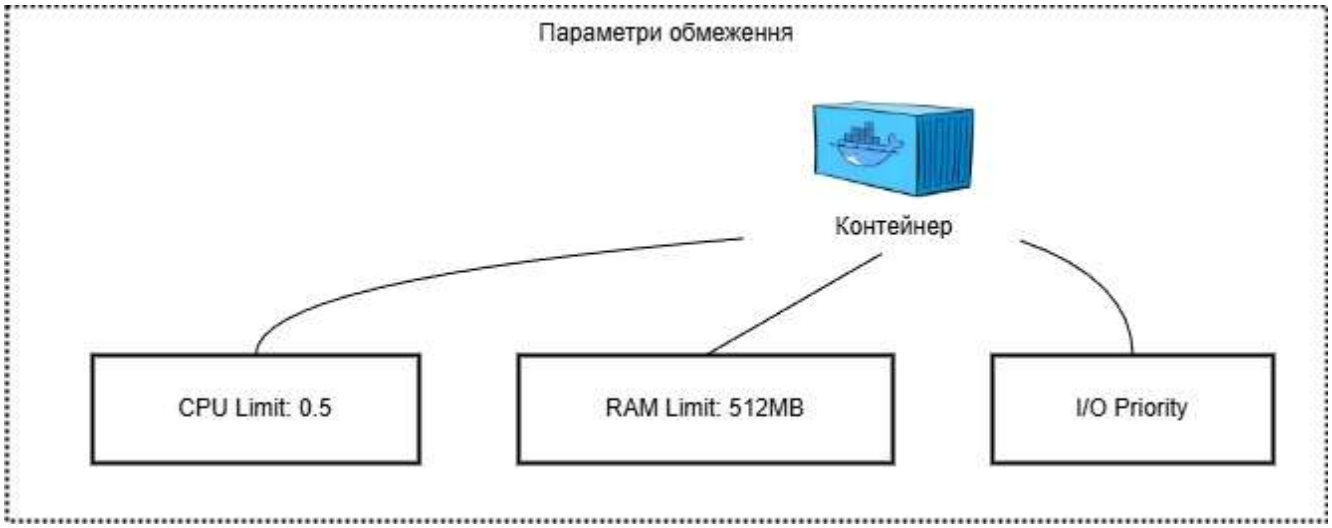


Рисунок 3.2 – Обмеження ресурсів контейнерів Docker

Додаткове підвищення стабільності досягається за рахунок використання механізмів обмеження процесорного часу для контейнерів, що дозволяє уникнути ситуацій, коли окремий процес монополізує ресурси CPU, а також застосування політик автоматичного перезапуску, які забезпечують відновлення роботи сервісів після збоїв без втручання адміністратора, ізоляція мережевого середовища через використання "macvlan" дозволяє зберігати чітке розмежування трафіку навіть при великій кількості вузлів, тоді як постійний моніторинг системних журналів забезпечує можливість своєчасного виявлення аномалій у поведінці контейнерів або мережевих процесів.

Сукупність зазначених підходів формує стабільне та передбачуване середовище функціонування IoT-сегменту на базі контейнеризації, у якому забезпечується контрольоване споживання ресурсів, відсутність критичних збоїв, пов’язаних із перевантаженням пам’яті або процесора, а також можливість масштабування без необхідності зміни загальної архітектури, у результаті чого досягається баланс між ефективністю використання ресурсів, продуктивністю системи та рівнем її надійності навіть при значному збільшенні кількості імітованих пристроїв.

3.2 Порівняльний аналіз із існуючими підходами до сегментації

Після побудови та оптимізації запропонованої архітектури захищеного IoT-сегменту доцільним стає проведення її порівняння з альтернативними підходами, що застосовуються у сучасних корпоративних мережах, оскільки такий аналіз дозволяє не лише обґрунтувати вибір використаних технологій, включаючи спеціалізований шлюз та контейнеризацію з мережевою ізоляцією, але й продемонструвати їхні переваги з точки зору безпеки, ефективності та масштабованості, при цьому особлива увага приділяється відмінностям між підходами до реалізації міжмережевого екранування, а також порівнянню контейнерної моделі імітації пристроїв із класичною повною віртуалізацією.

Переваги спеціалізованого шлюзу (pfSense) над вбудованими ОС-екранами. Розгляд архітектурних підходів до реалізації міжмережевого екрану дозволяє виділити дві базові моделі, одна з яких передбачає використання вбудованого екрану на рівні операційної системи, що функціонує безпосередньо на сервері та реалізується за допомогою інструментів фільтрації пакетів, інша ж базується на виділенні окремого мережевого вузла, який виконує роль шлюзу між сегментами та забезпечує централізовану обробку трафіку, у межах запропонованої архітектури використовується саме другий підхід, де міжмережевий екран функціонує як незалежний компонент інфраструктури, що фізично розмежовує площину передавання даних та площину керування, у результаті чого трафік клієнтів і контейнерів обробляється окремо від механізмів конфігурації правил доступу, що унеможливлює їх зміну з боку скомпрометованих вузлів.

Переваги такого підходу проявляються насамперед у рівні ізоляції управління, оскільки у випадку використання вбудованих засобів фільтрації на сервері компрометація операційної системи потенційно дозволяє змінювати правила доступу та відкривати несанкціоновані канали взаємодії між сегментами, тоді як винесення логіки фільтрації на окремий вузол усуває цю залежність і ускладнює обхід політик безпеки, додатково забезпечується централізоване

керування всією мережею через єдину точку конфігурації, що спрощує адміністрування та знижує ймовірність помилок при налаштуванні, особливо у випадках складних політик доступу, які включають велику кількість правил і винятків.

Окремою перевагою виступає розширене журналювання та можливість аудиту подій, де всі спроби доступу, як дозволені, так і заблоковані, фіксуються із зазначенням адрес, портів, протоколів та інтерфейсів, що значно спрощує аналіз мережевої активності та виявлення потенційних інцидентів безпеки, тоді як у випадку використання вбудованих механізмів операційної системи для досягнення аналогічного рівня контролю необхідне додаткове налаштування допоміжних інструментів, що ускладнює експлуатацію системи.

З точки зору стійкості до атак перевага також залишається за виділеним мережевим шлюзом, оскільки при компрометації окремого вузла, наприклад сервера з контейнерами, зловмисник не отримує прямого доступу до механізмів маршрутизації та фільтрації трафіку, що знаходяться поза межами цього вузла, у той час як при використанні host-based підходу така можливість потенційно існує, що створює додаткові ризики для всієї мережі.

Контейнеризація проти повної віртуалізації. Під час розгляду підходів до імітації або розгортання IoT-пристроїв у межах дослідницького стенду виділяються дві базові моделі, одна з яких передбачає використання повної віртуалізації, де кожен пристрій реалізується у вигляді окремої віртуальної машини з власною операційною системою та повною ізоляцією на рівні гіпервізора, інша ж ґрунтується на контейнеризації, де ізоляція досягається на рівні процесів із використанням спільного ядра операційної системи, що суттєво знижує накладні витрати на ресурси, у результаті чого виникає принципова різниця у споживанні оперативної пам’яті та обчислювальних потужностей, яка стає особливо помітною при масштабуванні середовища.

У випадку повної віртуалізації кожен імітований пристрій потребує виділення окремого обсягу оперативної пам’яті, який у мінімальній конфігурації становить сотні мегабайтів, що призводить до того, що навіть невелика кількість

вузлів формує значне навантаження на систему, наприклад п’ять віртуальних машин можуть споживати близько кількох гігабайтів оперативної пам’яті, при цьому частина ресурсів витрачається не лише на виконання прикладних процесів, але й на підтримку повноцінного середовища операційної системи для кожного вузла, тоді як контейнеризація дозволяє уникнути цих витрат за рахунок використання спільного ядра, у межах якого кожен контейнер запускається як ізольований процес, що значно зменшує обсяг необхідної пам’яті, при використанні легковагових образів споживання ресурсів одним контейнером може обмежуватися десятками мегабайтів, у результаті чого навіть декілька десятків контейнерів можуть функціонувати в межах обсягу пам’яті, співставного з однією або кількома віртуальними машинами, що забезпечує економію ресурсів на рівні понад дев’яносто відсотків у порівнянні з класичною віртуалізацією.

Додатковою перевагою контейнерного підходу виступає характер мережевої взаємодії, оскільки завдяки використанню спеціалізованого мережевого драйвера контейнери отримують власні IP-адреси та унікальні MAC-адреси, що дозволяє їм функціонувати як повноцінні вузли другого рівня мережевої моделі, у такій конфігурації кожен контейнер сприймається маршрутизатором як окремий фізичний пристрій, до якого застосовуються стандартні правила маршрутизації та контролю доступу без необхідності додаткових механізмів трансляції, що забезпечує коректну інтеграцію контейнерів у загальну мережеву топологію та дозволяє відтворити поведінку реальних IoT-пристроїв із високим ступенем точності [29].

Переваги контейнеризації також проявляються у швидкості розгортання та гнучкості конфігурації, оскільки створення нового вузла не потребує інсталяції окремої операційної системи, а зводиться до запуску нового контейнера, що може бути виконано за лічені секунди, масштабування середовища до десятків або навіть сотень вузлів відбувається без суттєвого збільшення навантаження на систему, при цьому зміни у конфігурації можуть вноситися динамічно без необхідності перезапуску всієї інфраструктури, що значно спрощує процес тестування різних сценаріїв та адаптації системи до нових умов.

Разом із тим контейнерний підхід має певні обмеження, пов’язані зі спільним використанням ядра операційної системи, що забезпечує нижчий рівень ізоляції у порівнянні з повною віртуалізацією, а також залежність від стану хост-системи, додаткові складнощі можуть виникати під час налаштування мережевих драйверів, зокрема у випадку використання macvlan, що вимагає більш глибокого розуміння мережевої моделі, однак у контексті дослідницького стенду та задач моделювання IoT-середовища такі обмеження не є критичними, оскільки пріоритетом виступає ефективність використання ресурсів та можливість швидкого масштабування (див. табл. 3.1).

Таблиця 3.1 – Порівняння контейнеризації та повної віртуалізації

Критерій	Docker containers	Virtual machines
Рівень віртуалізації	Рівень ОС	Рівень гіпервізора
Споживання RAM	Низьке (десятки МБ)	Високе (сотні МБ – ГБ)
Використання CPU	Менше накладних витрат	Вище навантаження
Швидкість запуску	Секунди	Хвилини
Масштабованість	Висока	Обмежена ресурсами
Ізоляція	Логічна	Повна
Окрема ОС	Не потрібна	Потрібна
Мережеві можливості	Власні IP/MAC через macvlan	Повноцінний мережевий стек
Гнучкість конфігурації	Висока	Середня
Реалістичність моделювання IoT	Висока	Висока
Складність адміністрування	Нижча	Вища
Доцільність для стенду	Оптимальна	Ресурсозатратна

Узагальнення порівняльного аналізу дозволяє зробити висновок про доцільність використання контейнеризації як основного інструменту імітації IoT-пристроїв у межах запропонованої архітектури, оскільки даний підхід забезпечує значно нижчі витрати ресурсів, високу швидкість розгортання, достатній рівень мережевої ізоляції та можливість гнучкого масштабування.

3.3. Перспективи модернізації та масштабування системи безпеки

Розроблений у межах дослідження віртуальний стенд формує базову, але водночас ефективну модель ізоляції IoT-сегменту в корпоративному середовищі, однак реальні інфраструктури підприємств характеризуються значно більшими масштабами, підвищеними вимогами до відмовостійкості та розширеними функціональними можливостями, у зв’язку з чим виникає необхідність подальшого розвитку запропонованої архітектури, що включає перехід від віртуального середовища до фізичних платформ, впровадження систем активного моніторингу та аналізу трафіку, підвищення рівня автоматизації управління політиками безпеки та масштабування сегментації на рівень повноцінної корпоративної мережі.

Міграція на апаратні платформи рівня підприємства. Одним із ключових напрямків модернізації виступає перенесення розробленої моделі у фізичну інфраструктуру, оскільки архітектура побудована на основі стандартних мережевих протоколів, що забезпечує її повну сумісність із реальним обладнанням без необхідності принципових змін у логіці роботи, у віртуальному середовищі ізольовані мережі реалізуються через внутрішні комутатори, тоді як у фізичній мережі їх аналогом виступають віртуальні локальні мережі з використанням механізму тегування трафіку відповідно до стандарту 802.1Q, де кожен сегмент отримує власний ідентифікатор та ізольований широкомовний домен, у такій моделі корпоративний сегмент і IoT-сегмент функціонують у різних VLAN із відповідною IP-адресацією, а маршрутизація між ними здійснюється через центральний шлюз, що забезпечує контроль доступу та застосування політик безпеки.

Фізична топологія передбачає наявність зовнішнього підключення до мережі Інтернет, після чого трафік надходить до апаратного маршрутизатора або міжмережевого екрану, який виконує функції фільтрації та маршрутизації, далі через транкове з’єднання передається до керованого комутатора, від якого формуються окремі сегменти для користувацьких пристроїв та IoT-обладнання, де

один сегмент обслуговує робочі станції співробітників, а інший — датчики, камери та інші периферійні пристрої, ізоляція між цими сегментами забезпечується на рівні комутатора та додатково контролюється на рівні маршрутизатора, що дозволяє реалізувати багаторівневу модель захисту.

Перехід до апаратних рішень також відкриває можливості використання спеціалізованих мережевих пристроїв, які мають апаратну оптимізацію обробки пакетів, що підвищує продуктивність та зменшує затримки, додатково забезпечується підтримка механізмів резервування, що дозволяє підвищити відмовостійкість системи за рахунок дублювання ключових компонентів, розділення площини керування та площини передавання даних дозволяє підвищити безпеку адміністрування та унеможливити несанкціонований доступ до конфігурації мережі, масштабованість такої інфраструктури забезпечується підтримкою великої кількості VLAN та можливістю обслуговування значної кількості пристроїв без втрати ефективності (див. рис. 3.4).

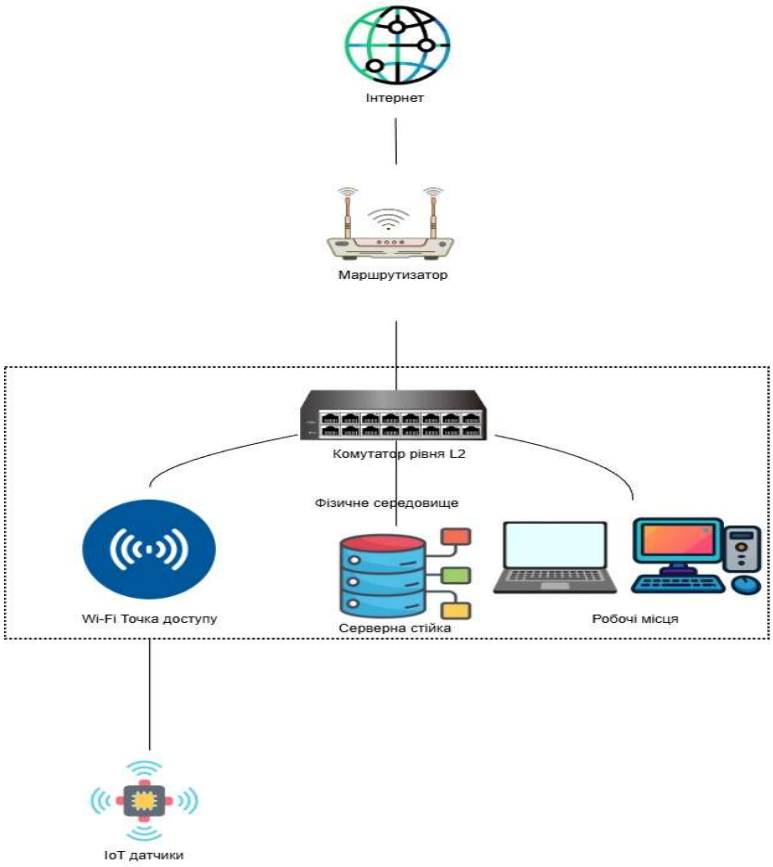


Рисунок 3.4 – Архітектура корпоративної мережі з фізичним обладнанням

Разом із перевагами впровадження фізичної інфраструктури виникають певні виклики, пов’язані з необхідністю придбання обладнання, ускладненням процесу конфігурації великих мереж та потребою у постійному адмініструванні і моніторингу, однак такі обмеження є типовими для корпоративного середовища і компенсуються значним підвищенням рівня безпеки, продуктивності та надійності системи.

Інтеграція систем виявлення вторгнень (IDS/IPS). Подальший розвиток архітектури логічно переходить від базового пасивного захисту до більш складних механізмів активного контролю трафіку, де початкова модель, що обмежується рівнями L3–L4 із фільтрацією за IP-адресами, портами та протоколами, доповнюється інструментами глибокого аналізу мережевої взаємодії, що дозволяє виявляти не лише явні порушення політик доступу, але й складніші атаки, які маскуються під легітимний трафік або використовують особливості прикладних протоколів, у результаті чого формується необхідність інтеграції систем IDS/IPS як складової частини мережевого екрану, здатної здійснювати як моніторинг, так і активне реагування на загрози (див. рис. 3.5).

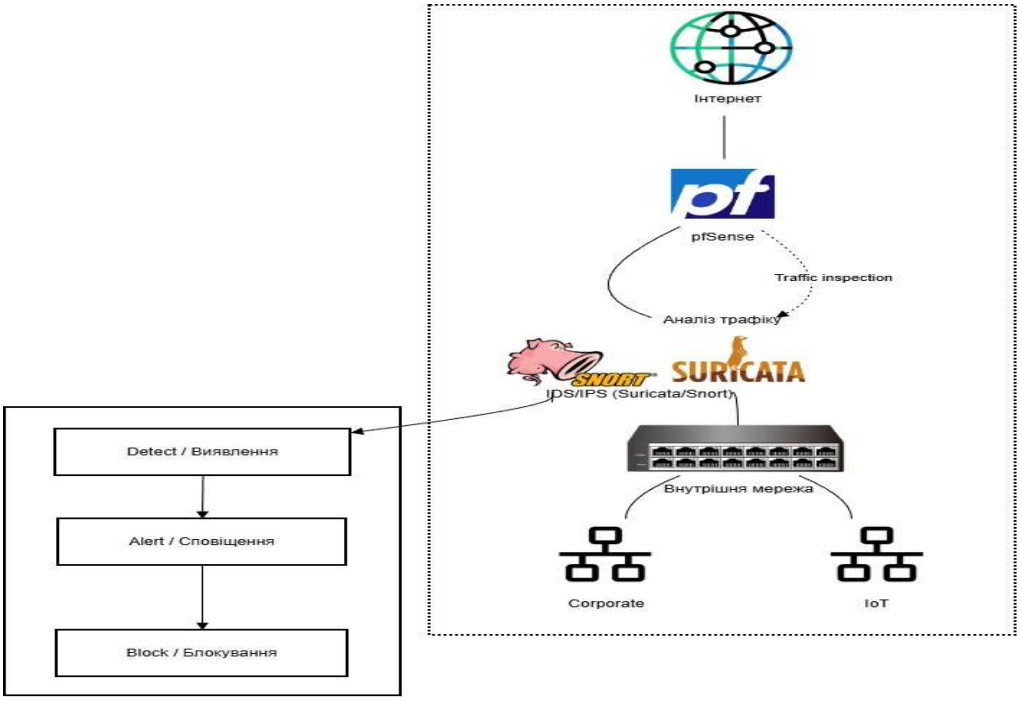


Рисунок 3.5 – Архітектура інтеграції IDS/IPS у pfSense

У межах реалізації подібного підходу можуть застосовуватися рішення, інтегровані безпосередньо у середовище маршрутизатора, що забезпечує централізований контроль, де сигнатурні та поведінкові механізми аналізу дозволяють обробляти трафік на рівні L7 із розбором протоколів, таких як HTTP, DNS або MQTT, при цьому аналіз не обмежується лише статичними правилами, а включає евристичні методи оцінки поведінки, що дозволяє виявляти аномалії навіть у випадках відсутності точного збігу із сигнатурою, і така логіка функціонування передбачає послідовність дій, у якій мережевий пакет проходить етап перевірки, зіставляється із базою сигнатур або поведінковими шаблонами, після чого формується рішення про його блокування або пропуск із відповідною фіксацією події у журналі [30f].

Реалізація подібного механізму дозволяє протидіяти типовим сценаріям атак, пов’язаним із експлуатацією вразливостей IoT-пристроїв, включаючи спроби несанкціонованого доступу через відкриті сервіси, сканування мережі, розповсюдження ботнетів або використання скомпрометованих вузлів для подальшого розширення атаки, при цьому кожна підозріла активність фіксується, аналізується та, за необхідності, блокується ще до досягнення цільового сегмента, що значно підвищує загальний рівень захищеності системи.

Переваги такого підходу проявляються у розширенні рівня захисту до прикладного рівня, можливості виявлення як відомих, так і нових типів атак, централізованому моніторингу подій та автоматизації реагування, що дозволяє зменшити залежність від ручного адміністрування, водночас впровадження IDS/IPS супроводжується певними обмеженнями, пов’язаними зі збільшенням навантаження на обчислювальні ресурси, необхідністю регулярного оновлення сигнатур та ймовірністю помилкових спрацювань, які можуть впливати на коректність обробки легітимного трафіку.

ВИСНОВКИ

При проєктуванні сформовано архітектурну модель із чітким розділенням мережевих зон у віртуальному середовищі. Топологія з логічними сегментами відтворила типову корпоративну інфраструктуру, де IoT-сегмент ізольовано від клієнтської мережі. Вибір програмного маршрутизатора та схема IP-адресації забезпечили базу для впровадження керованих політик доступу та тестування механізмів безпеки.

Під час розгортання створено віртуальне середовище з маршрутизатором, клієнтом та сервером імітації IoT. Контейнеризація дозволила ефективно запустити незалежні вузли в межах однієї ОС із повноцінною мережевою взаємодією. Налаштування мережевого стеку, зокрема використання macvlan, забезпечило коректну адресацію та ізоляцію контейнерів на другому рівні моделі OSI.

Реалізація маршрутизації та політик екрану підтвердила ефективність принципу мінімальних привілеїв. Забезпечено доступ корпоративного сегменту до зовнішніх ресурсів із повним блокуванням взаємодії з IoT. Тестування довело відповідність вимогам: наявність доступу до Інтернету для клієнтів за відсутності прямого обміну трафіком між сегментами, що підтверджено результатами ICMP-запитів та журналами міжмережевого екрану.

Аналіз визначив напрями оптимізації системи, зокрема підвищення швидкості обробки правил та зменшення споживання ресурсів. Порівняння з аналогами підтвердило переваги використання окремого шлюзу безпеки та контейнеризації. Перспективи включають масштабування до фізичної інфраструктури з VLAN та апаратними маршрутизаторами, а також інтеграцію систем виявлення вторгнень для глибокого аналізу трафіку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. IoT Security Foundation. IoT Security Compliance Framework. URL: https://www.iotsecurityfoundation.org (дата звернення: 25.02.2026).
2. OWASP. Internet of Things Top 10. URL: https://owasp.org/www-project-internet-of-things (дата звернення: 28.02.2026).
3. Cisco. IoT Threat Defense. URL: https://www.cisco.com (дата звернення: 03.03.2026).
4. ENISA. Baseline Security Recommendations for IoT. URL: https://www.enisa.europa.eu (дата звернення: 07.03.2026).
5. NIST. Guidelines for Securing Wireless Networks. URL: https://nvlpubs.nist.gov (дата звернення: 10.03.2026).
6. IEEE. VLAN Technologies Overview. URL: https://ieeexplore.ieee.org (дата звернення: 14.03.2026).
7. Cisco. VLAN Configuration Guide. URL: https://www.cisco.com (дата звернення: 18.03.2026).
8. Juniper Networks. Layer 3 Routing Basics. URL: https://www.juniper.net (дата звернення: 21.03.2026).
9. Cloudflare. What is Routing? URL: https://www.cloudflare.com/learning/network-layer/what-is-routing (дата звернення: 25.03.2026).
10. NIST. Access Control Systems. URL: https://csrc.nist.gov (дата звернення: 29.03.2026).
11. Palo Alto Networks. What is a Firewall? URL: https://www.paloaltonetworks.com (дата звернення: 01.04.2026).
12. Fortinet. ACL Explained. URL: https://www.fortinet.com/resources/cyberglossary/acl (дата звернення: 04.04.2026).
13. Microsoft. Principle of Least Privilege. URL: https://learn.microsoft.com (дата звернення: 08.04.2026).
14. VMware. Virtual Networking Concepts. URL: https://docs.vmware.com (дата звернення: 12.04.2026).
15. Red Hat. What is Containerization? URL: https://www.redhat.com/en/topics/containers (дата звернення: 15.04.2026).
16. Docker. Networking Overview. URL: https://docs.docker.com/network (дата звернення: 19.04.2026).
17. ISO/IEC 27001 Overview. URL: https://www.iso.org/isoiec-27001-information-security.html (дата звернення: 22.04.2026).
18. NIST. Security Metrics Guide. URL: https://csrc.nist.gov/publications (дата звернення: 26.04.2026).
19. Microsoft. System Requirements for Virtualization. URL: https://learn.microsoft.com (дата звернення: 29.04.2026).

20. pfSense Documentation. URL: https://docs.netgate.com/pfsense (дата звернення: 01.05.2026).
21. VMware Workstation Network Editor Guide. URL: https://www.vmware.com (дата звернення: 02.05.2026).
22. Netgate. pfSense Interface Configuration. URL: https://www.netgate.com (дата звернення: 03.05.2026).
23. pfSense Firewall Rules Documentation. URL: https://docs.netgate.com/pfsense/en/latest/firewall (дата звернення: 03.05.2026).
24. Docker Compose Reference. URL: https://docs.docker.com/compose (дата звернення: 04.05.2026).
25. Docker Macvlan Network Driver. URL: https://docs.docker.com/network/macvlan (дата звернення: 04.05.2026).
26. Cisco. NAT Fundamentals. URL: https://www.cisco.com (дата звернення: 04.05.2026).
27. pfSense Logging and Monitoring. URL: https://docs.netgate.com (дата звернення: 05.05.2026).
28. Netgate. Firewall Optimization Techniques. URL: https://www.netgate.com (дата звернення: 05.05.2026).
29. Scale Computing. Virtualization vs Containerization. URL: https://www.scalecomputing.com/resources/whats-the-difference-between-virtualization-vs-containerization (дата звернення: 05.05.2026).
30. Snort Project. Network Intrusion Detection System. URL: https://www.snort.org (дата звернення: 05.05.2026).
31. Лемешко А.В., Антоненко А.В., Петрик А.В. (2023) Нейроморфні системи як інструмент реалізації штучного інтелекту. Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки. Том 34 (73) № 3 2023, 175-183. https://doi.org/10.32782/2663-5941/2023.3.1/28
32. Лемешко А. В., Антоненко А. В., Балвак А. А., & Новіченко Є. О. (2023). Актуальні засади створення алгоритмів обробки інформації для логістичних центрів. Таврійський науковий вісник. Серія: Технічні науки, (1), 25-32. https://doi.org/10.32851/tnv-tech.2023.1.3
33. Гакман Д. В. & Антоненко А. В. (2023). Особливості розробки та використання фреймворків для автоматизованого тестування. Таврійський науковий вісник. Серія: Технічні науки, (2), 21-32. https://doi.org/10.32782/tnv-tech.2023.2.3
34. Сорочинський О. О., Гунько О. С., Антоненко А. В., Шульженко К. Ю., & Христенко Д. О. (2023). Моделювання та проектування приватного мережево-серверного середовища з використанням технології opennebula. Таврійський науковий вісник. Серія: Технічні науки, (3), 3-12. https://doi.org/10.32782/tnv-tech.2023.3.1
35. Лемешко А., Добровольський О., Антоненко А., Завадський В. Оптимізація часу реагування веб серверу apache. Вісник Хмельницького національного університету, Том 1, №2, 2023 (319). 189-201 https://www.doi.org/10.31891/2307-5732-2023-319-1-189-201

36. Лемешко А., Антоненко А., Цвик О. Аналіз і особливості програмного забезпечення для контролю трафіку. Вісник Хмельницького національного університету, Том 1, №3, 2023 (321). 64-68 https://www.doi.org/10.31891/2307-5732-2023-321-3-64-68

37. Лемешко А., Кувік Н., Антоненко А., Гнядий В. Дослідження мережевих загроз для забезпечення максимального захисту даних та інфраструктури. Вісник Хмельницького національного університету, Том 1, №3, 2023 (321). 64-68 https://www.doi.org/10.31891/2307-5732-2023-321-3-120-127

38. Пахомов М., Калита Т., Антоненко А., Галета В. Використання штучного інтелекту в автоматизованих системах. Вісник Хмельницького національного університету, №4, 2023 (323). 11-20. https://www.doi.org/10.31891/2307-5732-2023-323-4-11-20

39. Поперешняк С.В., Вєчерковська А.С., Житкевич О.Б., Антоненко А.В. Моніторинг руху об'єктів засобами геоінформаційних систем. Вісник Хмельницького національного університету, №6, 2023 (329) 312-320. https://www.doi.org/10.31891/2307-5732-2023-329-6-312-320

40. Поперешняк С. В., Вєчерковська А. С., Хільченко М. Ю., & Антоненко А. В. (2024). Хмарні технології як сервіси для оптимізації процесів адміністрування. Таврійський науковий вісник. Серія: Технічні науки, (6), 54-63. https://doi.org/10.32782/tnv-tech.2023.6.7

41. Поперешняк С. В., Куценко М. О., & Антоненко А. В. (2024). Система надання рекомендацій щодо вивчення технологій з веб-розробки. Таврійський науковий вісник. Серія: Технічні науки, (6), 64-71. https://doi.org/10.32782/tnv-tech.2023.6.8

42. Бондаренко Є. С., Каньшин К. В., Антоненко А. В., Степанчук В. І., & Ланевський Л. А. (2023). Класифікація та особливості збору даних у системах підтримки прийняття управлінських рішень. Таврійський науковий вісник. Серія: Технічні науки, (4), 23-31. https://doi.org/10.32782/tnv-tech.2023.4.3

43. Бенедіко І. В., Вічкарук А. І., Антоненко А. В., Лисенко К. В., & Сижко О. Ю. (2023). Класифікації моделей застосування машинного навчання у кібербезпеці. Таврійський науковий вісник. Серія: Технічні науки, (4), 11-22. https://doi.org/10.32782/tnv-tech.2023.4.2

44. Зайцев Є. О., Березниченко В. О., Антоненко А. В., Панчик М. В., & Тонких О. Г. (2024). Впровадження технологій іот в системах електропостачання промислових підприємств. Таврійський науковий вісник. Серія: Технічні науки, (5), 23-32. https://doi.org/10.32782/tnv-tech.2023.5.3

45. Кірюшин В. О., Лемешко А. В., Антоненко А. В., Рябоконь О. О., & Макаренко Є. О. (2024). Особливості функціональності dns-сервера та dhcp-сервера. Таврійський науковий вісник. Серія: Технічні науки, (5), 55-64. https://doi.org/10.32782/tnv-tech.2023.5.6

46. Шрам М.М., Замковий О.Ю., Антоненко А.В., Орлюк Х.В., Дутко П.М. Особливості побудови масиву даних на основі нейромережі у сфері інтернету речей для готельно-ресторанного бізнесу. Вісник Хмельницького національного університету, Том 1, №5, 2023 (325), 252-257. http://journals.khnu.km.ua/vestnik/wp-content/uploads/2023/11/325-n5-1-252-257.pdf

47. Поплавський Д. В., Голубенко О. І., Антоненко А.В., Кондратюк Б. О., Верко Р. А. Впровадження моделі розгортання мереж мобільного зв'язку 6-го покоління за допомогою безпілотників. Вісник Хмельницького національного університету, №6, 2023 (329), 49-55. http://journals.khnu.km.ua/vestnik/wp-content/uploads/2024/01/329-ts-2023-n6-49-55.pdf
https://doi.org/10.31548/humanhealth.4.2024.37

48. Зайцев Є.О., Березниченко В.О., Антоненко А.В., Закусило С.А. (2022). SMART засоби визначення аварійних станів у розподільних електричних мережах міст. Таврійський науковий вісник. Серія: Технічні науки, (5), 3-12. https://doi.org/10.32851/tnv-tech.2022.5.1

49. Буряк, М. С., Цвик, О. С., Антоненко, А. В., Бурачинський, А. Ю., Балвак, А. А., & Занфіров, Р. Р. (2024). Особливості розгортання мереж мобільного зв'язку за допомогою бпла. Таврійський науковий вісник. Серія: Технічні науки, (3), 3-12. URL: https://doi.org/10.32782/tnv-tech.2024.3.1

50. Балвак, А. А., Лемешко, А. В., Антоненко, А. В., Зіняр, Д. А., Бурачинський, А. Ю., & Приходько, А. П. (2024). Обробка та аналіз даних на прикладі набору spambase з використанням бібліотек для машинного навчання. Таврійський науковий вісник. Серія: Технічні науки, (2), 3-20. https://doi.org/10.32782/tnv-tech.2024.2.1

51. Твердохліб, А. О., Востріков, С. О., Антоненко, А. В., Бегліцов, С. В., Гергель, О. Г., & Бобков, Р. А. (2024). Оптимізація роботи бездротових мереж з використанням inet framework. Таврійський науковий вісник. Серія: Технічні науки, (5), 11-24. https://doi.org/10.32782/tnv-tech.2024.5.2

52. Твердохліб, А. О., & Антоненко, А. В. (2024). Особливості розподіленого брокеру trinity на основі технології блокчейн. Таврійський науковий вісник. Серія: Технічні науки, (6), 130-138. URL: https://doi.org/10.32782/tnv-tech.2024.6.14

53. Солобаєв, С. Г., Востріков, С. О., Антоненко, А. В., Балвак, А. А., Мішкур, Ю. В., & Приходько, А. П. (2024). Аспекти використання нейронних мереж для покращення якості зображень. Таврійський науковий вісник. Серія: Технічні науки, (6), 3-10. URL: https://doi.org/10.32782/tnv-tech.2024.6.1

54. Балвак, А. А., Цвик, О. С., Антоненко, А. В., Ємелін, Д. М., & Гришковець, Є. П. (2024). Інноваційні методи відображення інформації у веб-браузерах. Таврійський науковий вісник. Серія: Технічні науки, (1), 3-11. https://doi.org/10.32782/tnv-tech.2024.1.1

55. Востріков, С. О., Бурачинський, А. Ю., Антоненко, А. В., Твердохліб, А. О., Балвак, А. А., & Слободян, О. А. (2024). Особливості автоматизованого тестування з використанням фреймворків. Таврійський науковий вісник. Серія: Технічні науки, (4), 3-14. https://doi.org/10.32782/tnv-tech.2024.4.1

56. Мішкур, Ю. В., Сольський, Д. Я., Антоненко, А. В., Солобаєв, С. Г., Подуран, Д. В., & Сарафанюк, Р. О. (2024). Оптимізація продуктивності веб-сервера Apache. Таврійський науковий вісник. Серія: Технічні науки, (4), 15-30. https://doi.org/10.32782/tnv-tech.2024.4.2

57. Бурачинський А., Сольський D., Антоненко, А.,Твердохліб А., Мішкур Y., & Зіняр D. (2024). Аспекти застосування нейронних мереж для криптографії.

Measuring and computing devices in technological processes, (4), 394–400. https://doi.org/10.31891/2219-9365-2024-80-47

58. Мішкур, Ю., Твердохліб, А., Востріков, Антоненко, А., С., & Балвак, А. (2024). Нейромережі в мистецтві як іструмент графічного дизайну. Herald of Khmelnytskyi National University. Technical Sciences, 345(6(2), 95-101. https://doi.org/10.31891/2307-5732-2024-345-6-14

59. Солобаєв, С. Г., Востріков, С. О., Антоненко, А. В., Ткаченко, О. В., Ходосов, А. О., & Остапенко, О. С. (2025). Використання нейронних мереж у прогнозуванні безпеки мережі. Таврійський науковий вісник. Серія: Технічні науки, (2), 3-10. https://doi.org/10.32782/tnv-tech.2025.2.1

60. Черевик, О. В., Востріков, С. О., Антоненко, А. В., Мішкур, Ю. В., Балвак, А. А., & Дзісяк, В. Г. (2025). Оптимізація мережевої інфраструктури для забезпечення відмовостійкості. Таврійський науковий вісник. Серія: Технічні науки, (3), 3-13. https://doi.org/10.32782/tnv-tech.2025.3.1

61. Лемешко А. В., Антоненко А. В., Слюсар В. О. (2023) Технології хмарних обчислень та динаміка їх розвитку. ITSynergy, (1), 89–108. (IndexCopernicus) https://doi.org/10.53920/ITS-2023-1-6

62. Лемешко А. В., Антоненко А. В., Шрам М. М. (2023) Моделювання безпроводових мереж у середовищі OMNET++ з використанням Inet Framework. ITSynergy, (1), 37–59. (IndexCopernicus) https://doi.org/10.53920/ITS-2023-1-3

63. Лемешко А. В., Антоненко А. В., Матвійчук О. М. (2023). Управління трафіком в гібридній програмно-визначеній мережі. ITSynergy, (1), 98–114 (IndexCopernicus) https://doi.org/10.53920/ITS-2023-2-7 Sciences, (3), 3-8. https://doi.org/10.21303/2504-5695.2022.002605

64. Антоненко А., & Твердохліб А. (2025). Використання смарт-контрактів та технології блокчейн для підвищення ефективності розподілених брокерних систем в умовах високих навантажень. Measuring and computing devices in technological processes, (3), 467–473. https://doi.org/10.31891/2219-9365-2025-83-58

65. Buriak, M. S., Vostrikov, S. O., Antonenko, A. V., Balvak, A. A., Korotin, D. S., & Myronenko, R. O. (2025). Можливості підвищення ефективності мереж в системах клієнт- сервер. Таврійський науковий вісник. Серія: Технічні науки, 1(4), 3-14. https://doi.org/10.32782/tnv-tech.2025.4.1.1

66. Сольський D., Солобаєв S., Антоненко, А., Чечик S., & Черевик О. (2025). Використання бібліотеки scikit-learn у методах класифікації машинного навчання. Measuring and computing devices in technological processes, (1), 468–472. https://doi.org/10.31891/2219-9365-2025-81-58

67. Zaitsev I., Golubenko O., Tkachenko O., Pidmohylnyi O., Antonenko A. (2023). Exploring advanced hypothesis generation in astronomy through the implementation of a mathematical model of linguistic neural networks. CEUR Workshop Proceedings, 2023, 3687, 121–128 https://ceur-ws.org/Vol-3687/Short_5.pdf

68. Ткаченко, О., Голубенко, О., Власенко, В., & Антоненко, А. (2025). Методи оптимізації використання оперативної пам’яті в процесах підвищення швидкодії комп’ютерних систем. Вимірювальна та обчислювальна техніка в технологічних процесах, 84(4), 465–472. https://doi.org/10.31891/2219-9365-2025-84-57

69. Завгородня, Г., Завгородній, В., Голубенко, О., & Антоненко, А. (2025). Means of ensuring scalability and autonomy of an adaptive content generation system. Technologies and Engineering, 26(6), 21–31. https://doi.org/10.30857/2786-5371.2025.6.2

70. Golubenko, O., Onysko, A., Antonenko, A., Chechyk, S., & Vostrikov, S. (2025). Prospects of using AI for automated software testing. Technologies and Engineering, 26(5), 9–20. https://doi.org/10.30857/2786-5371.2025.5.1

Додаток А

Конфігураційний файл docker-compose.yml

```yaml
version: '3.8'

services:
  iot-sensor-1:
    image: alpine
    command: tail -f /dev/null
    networks:
      iot_macvlan:
        ipv4_address: 10.20.20.11

  iot-sensor-2:
    image: alpine
    command: tail -f /dev/null
    networks:
      iot_macvlan:
        ipv4_address: 10.20.20.12

  iot-sensor-3:
    image: alpine
    command: tail -f /dev/null
    networks:
      iot_macvlan:
        ipv4_address: 10.20.20.13

  iot-sensor-4:
    image: alpine
    command: tail -f /dev/null
    networks:
      iot_macvlan:
        ipv4_address: 10.20.20.14

  iot-sensor-5:
    image: alpine
    command: tail -f /dev/null
    networks:
      iot_macvlan:
        ipv4_address: 10.20.20.15

networks:
  iot_macvlan:
    driver: macvlan
    driver_opts:
      parent: ens33
    ipam:
      config:
        - subnet: 10.20.20.0/24
          gateway: 10.20.20.1
```

Додаток Б

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра комп'ютерної інженерії



КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавра

НА ТЕМУ: «МЕХАНІЗМИ ЗАХИСТУ ІОТ-ПРИСТРОЇВ У КОРПОРАТИВНІЙ ІНФРАСТРУКТУРІ»

Виконав студент групи КІД-41
Лабза Ілля Григорович
Керівник кваліфікаційної роботи:
Антоненко Артем Васильович,
канд. техн. наук, доц.

Київ – 2026

Мета, об’єкт, предмет дослідження

Мета роботи – мінімізація ризиків несанкціонованого доступу та поширення загроз у корпоративній мережі за рахунок використання принципів сегментації трафіку, міжмережевого екранування та ізоляції IoT-пристроїв

Об’єкт дослідження – процеси функціонування та взаємодії IoT-пристроїв у корпоративній мережевій інфраструктурі

Предмет дослідження – засоби та технології побудови захищених сегментів мережі, зокрема механізми VLAN-сегментації, маршрутизації, ACL та міжмережевих екранів

Актуальність роботи

У роботі досліджено теоретичні аспекти безпеки IoT, методи мережевої ізоляції на основі VLAN та управління доступом через ACL. Розглянуто принципи сегментації трафіку та моделювання мереж у віртуальних середовищах із використанням гіпервізорів і технологій контейнеризації.

Практичну частину присвячено реалізації захищеного сегмента на базі програмного маршрутизатора pfSense та імітації IoT-пристроїв у контейнерах Docker. Налаштовано політики безпеки, проведено тестування ізоляції та оптимізовано роботу міжмережевого екрана з визначенням перспектив інтеграції систем IDS/IPS.

Архітектура запропонованого рішення

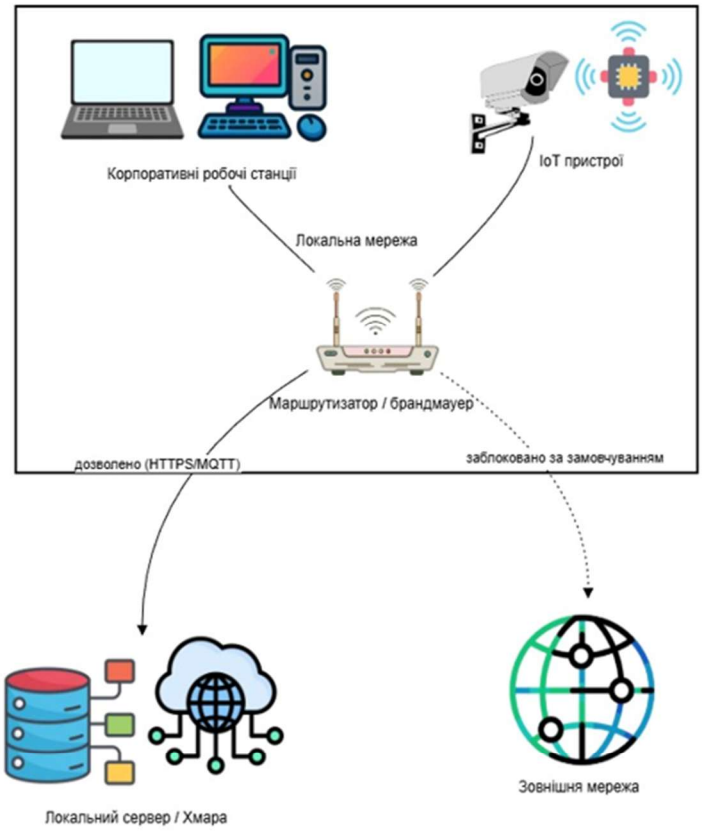


Рис. 2.1 – Концептуальна архітектура захищеного IoT-сегменту

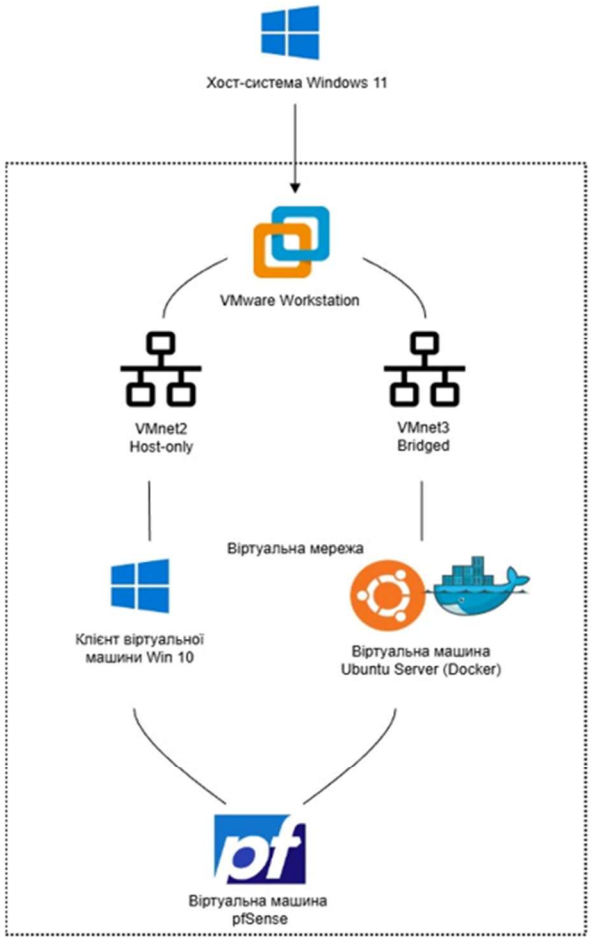


Рис. 2.2 – Мережева топологія віртуального стенду

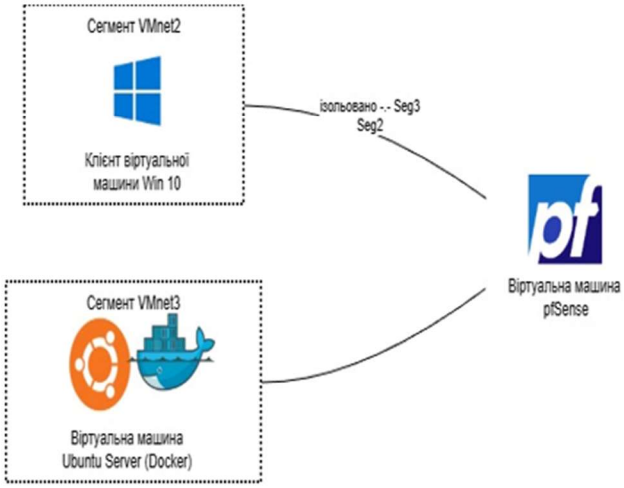


Рис. 2.3 – Логічне розділення сегментів VMnet2 та VMnet3

Схема адресації та сегментації

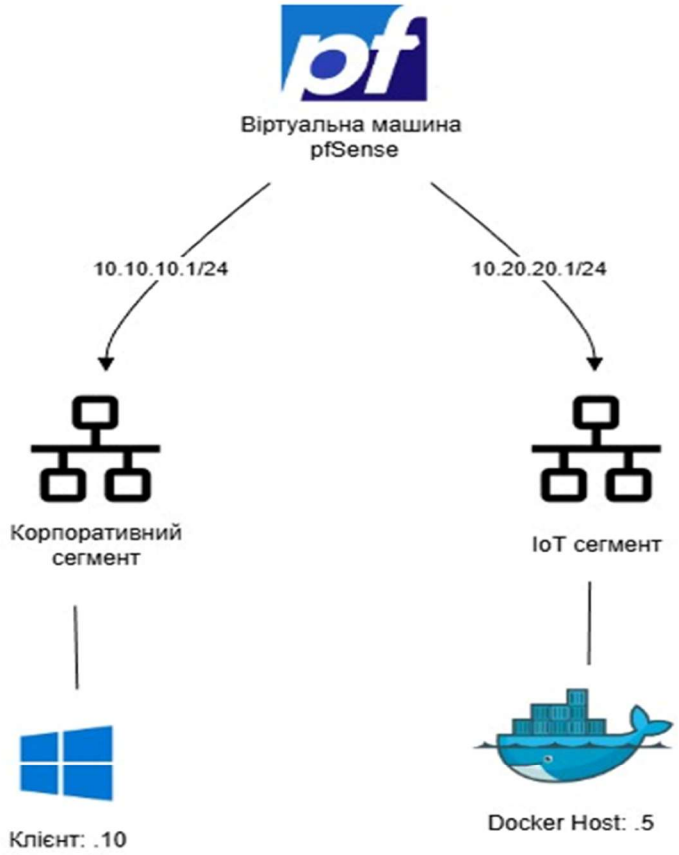


Рис. 2.4 – Схема IP-адресації мережі

Табл. 2.1 – Адресація та логічна сегментація віртуального середовища

Сегмент	VMnet	IP-мережа	Gateway	Основні вузли	Призначення
WAN	VMnet8	192.168.x.x	DHCP	pfSense WAN	Доступ до Internet через NAT
Corporate	VMnet2	10.10.10.0/24	10.10.10.1	Windows 10 (10.10.10.10)	Корпоративний сегмент
IoT	VMnet3	10.20.20.0/24	10.20.20.1	Ubuntu Server (10.20.20.5), Docker containers (10.20.20.11–15)	Ізольований сегмент IoT-пристроїв

Практична реалізація середовища

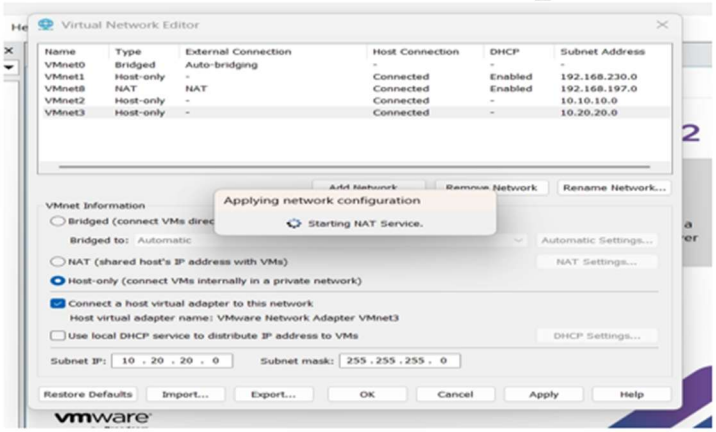


Рис. 2.5 – Налаштування мережевих адаптерів у VMware

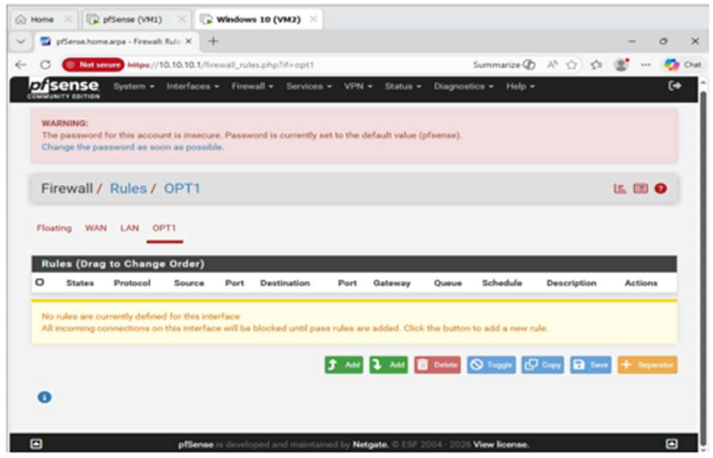


Рис. 2.7 – Інтерфейс налаштування правил pfSense

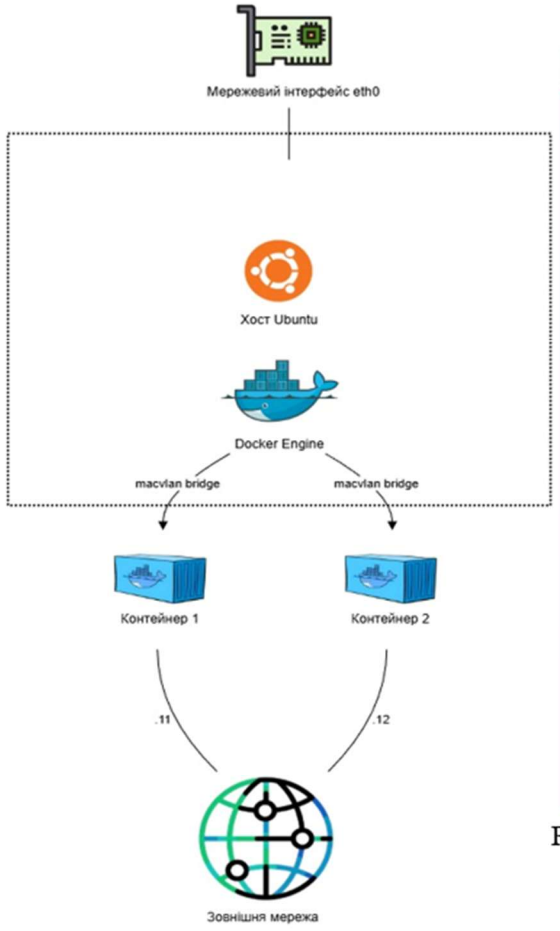


Рис. 2.9 – Схема мережі "macvlan" у Docker

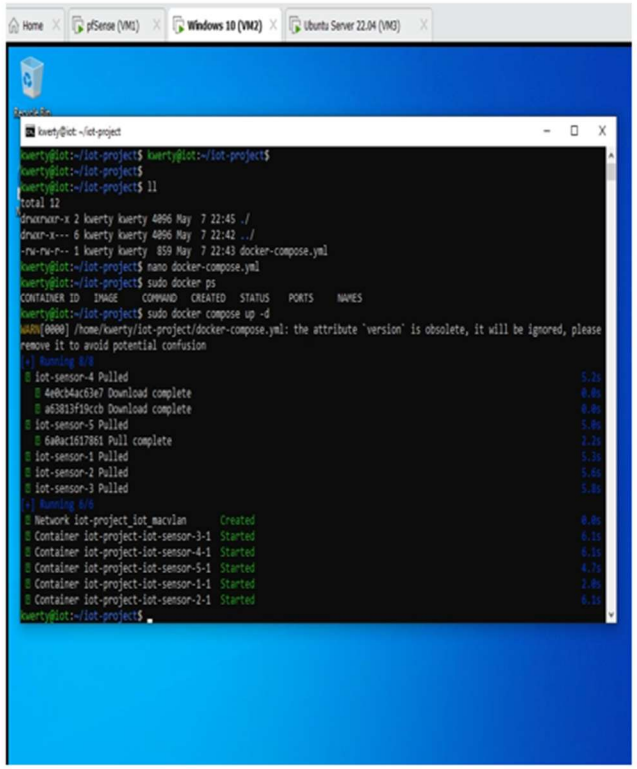


Рис. 2.10 – Результат запуску контейнерів у терміналі

Алгоритми фільтрації та безпеки

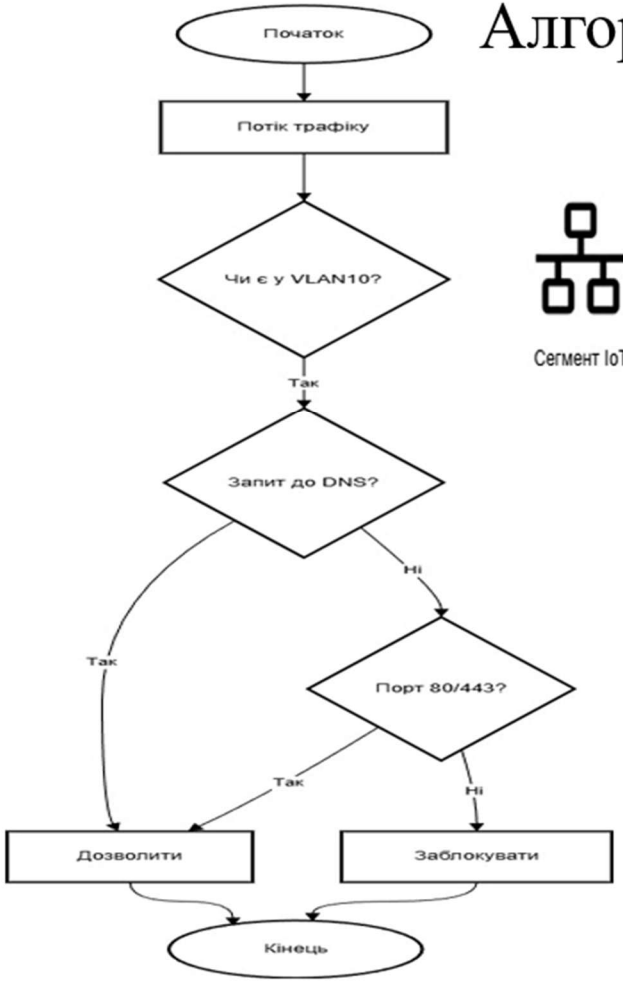


Рис. 2.11 – Алгоритм обробки трафіку корпоративного сегменту

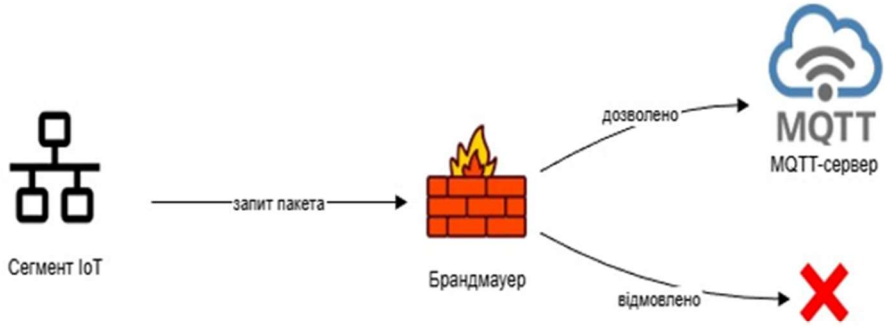


Рис. 1.6 – Модель "Default Deny" у мережевій безпеці

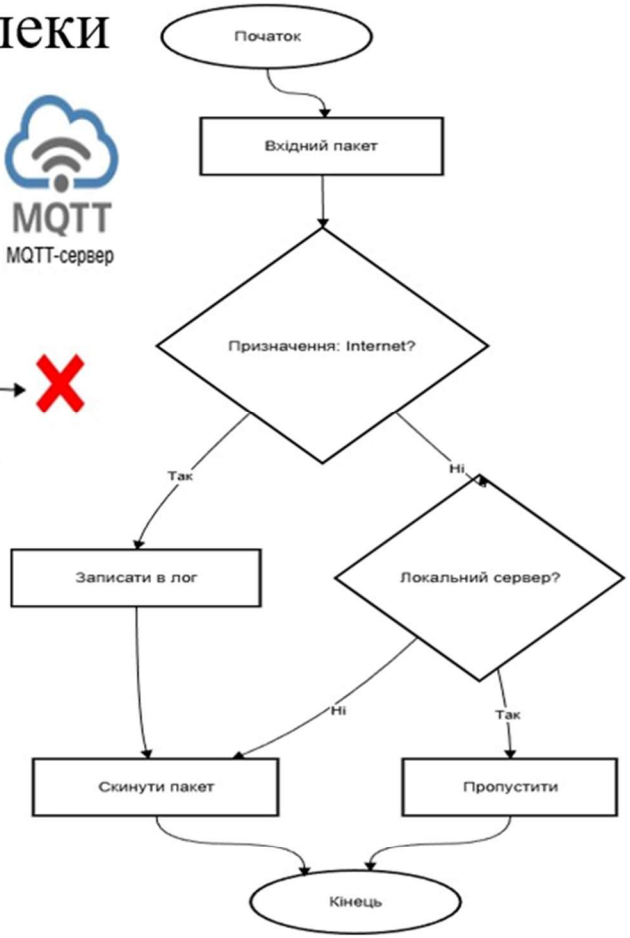


Рис. 2.12 – Алгоритм блокування трафіку IoT

Тестування та підтвердження ізоляції



Рис. 2.13 – Результат ping-запиту до Інтернету

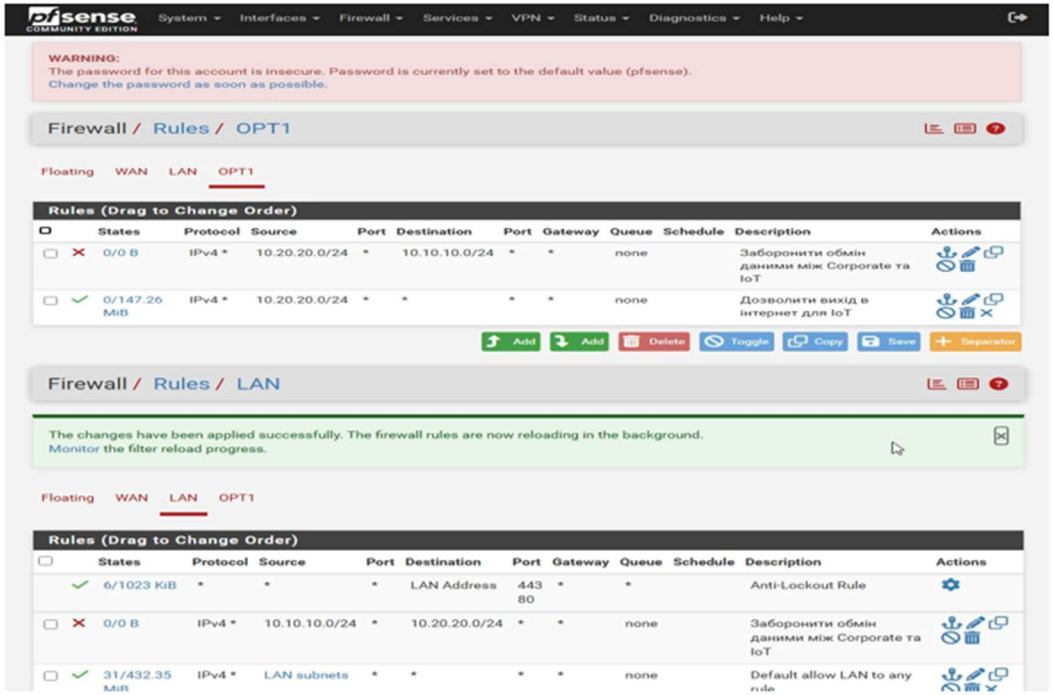


Рис. 2.15 – Результат блокування між сегментами

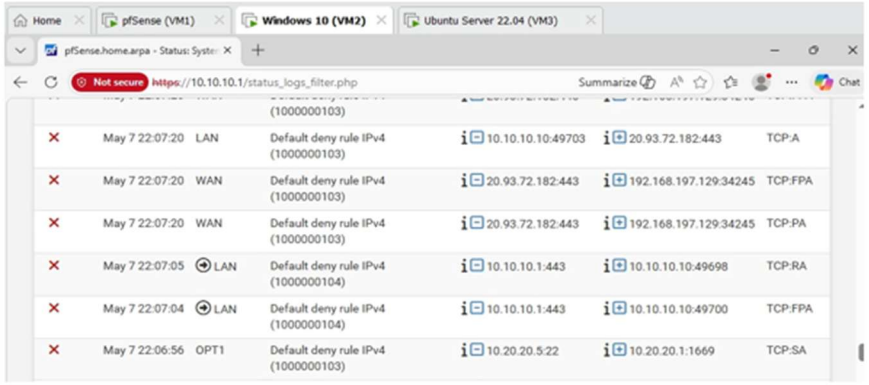


Рис. 2.17 – Логи pfSense (Firewall Logs)

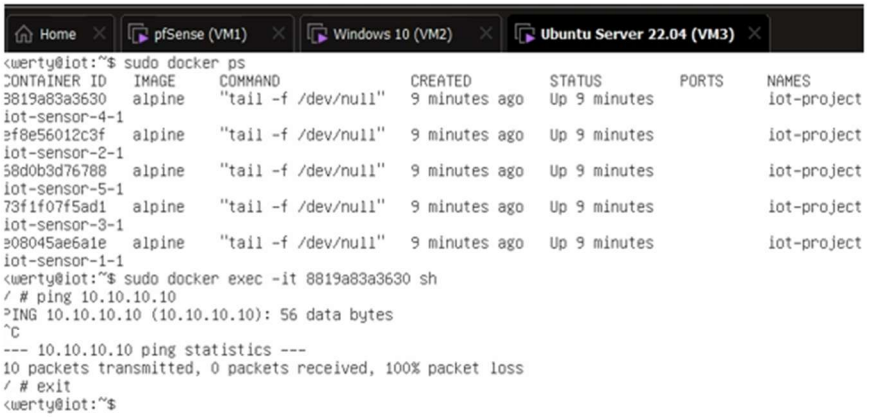


Рис. 2.16 – Спроба доступу IoT до корпоративної мережі

Оптимізація та продуктивність

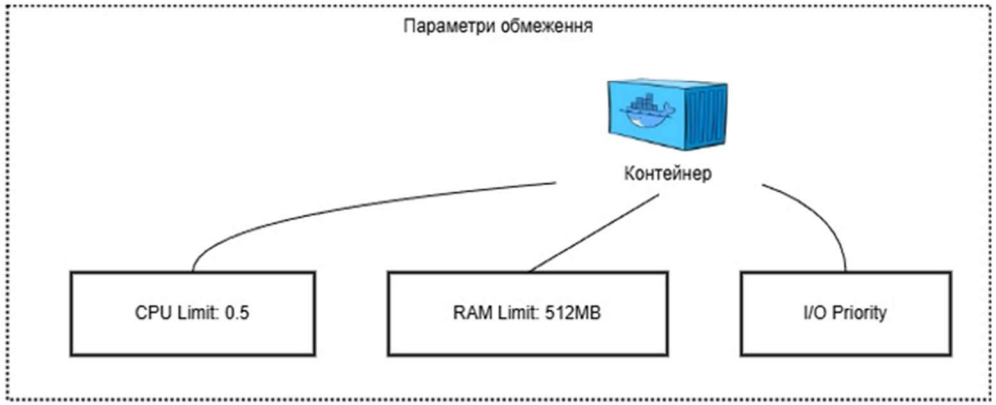


Рис. 3.2 – Обмеження ресурсів контейнерів Docker

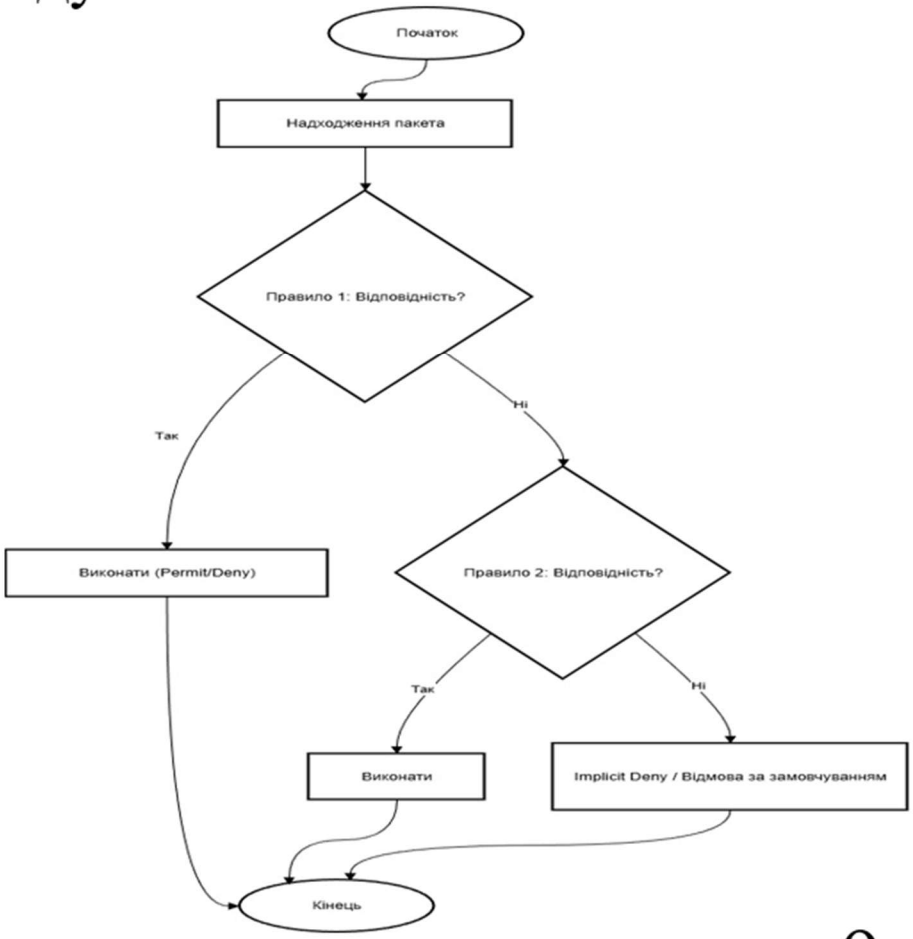


Рис. 3.1 – Порядок обробки правил ACL

Порівняння підходів

Табл. 3.1 – Порівняння контейнеризації та повної віртуалізації

Критерій	Docker containers	Virtual machines
Рівень віртуалізації	Рівень ОС	Рівень гіпервізора
Споживання RAM	Низьке (десятки МБ)	Високе (сотні МБ – ГБ)
Використання CPU	Менше накладних витрат	Вище навантаження
Швидкість запуску	Секунди	Хвилини
Масштабованість	Висока	Обмежена ресурсами
Ізоляція	Логічна	Повна
Окрема ОС	Не потрібна	Потрібна
Мережеві можливості	Власні IP/MAC через macvlan	Повноцінний мережевий стек
Гнучкість конфігурації	Висока	Середня
Реалістичність моделювання IoT	Висока	Висока
Складність адміністрування	Нижча	Вища
Доцільність для стенду	Оптимальна	Ресурсозатратна

Перспективи розвитку

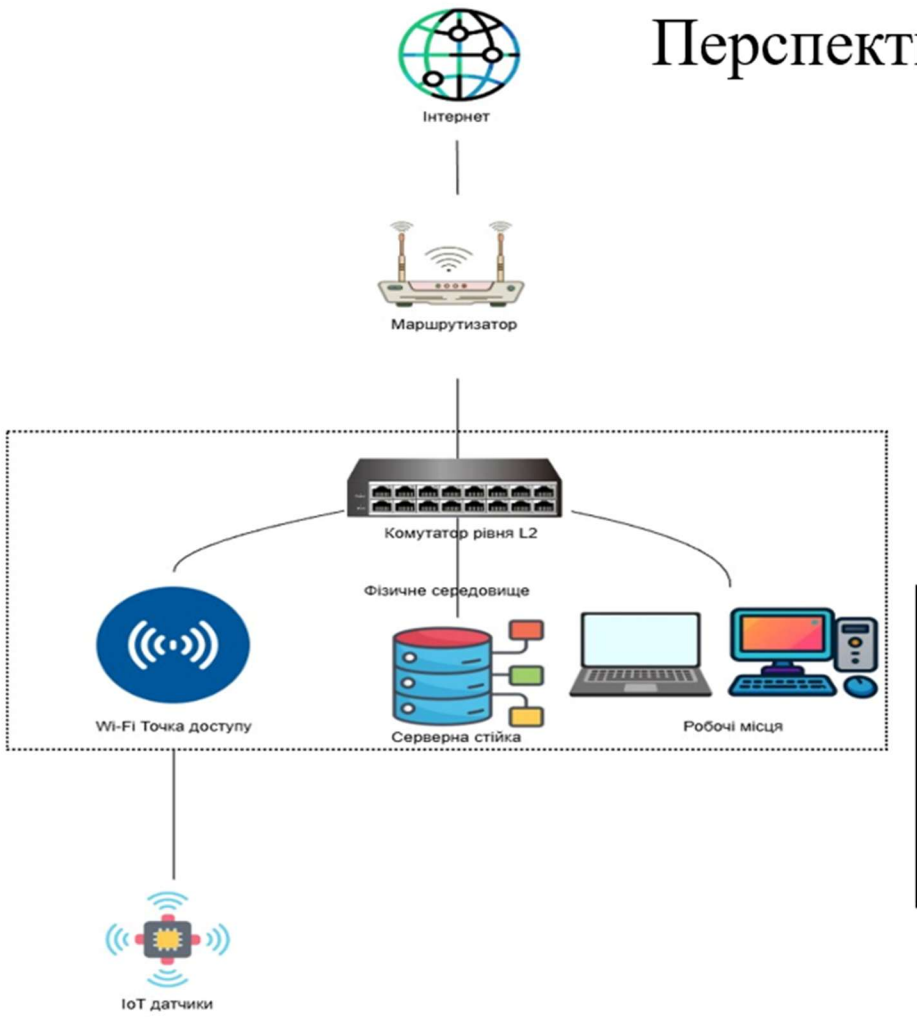


Рис. 3.4 – Архітектура корпоративної мережі з фізичним обладнанням

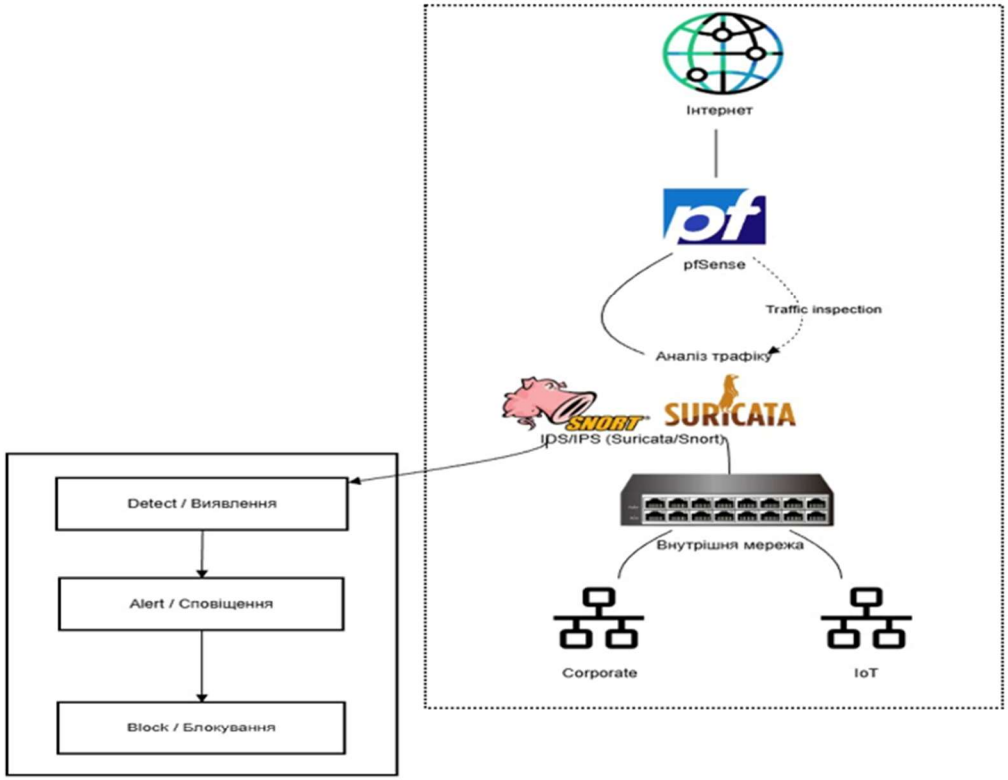


Рис. 3.5 – Архітектура інтеграції IDS/IPS у pfSense

Висновки

У ході роботи було розроблено та розгорнуто архітектурну модель захищеного середовища, що забезпечує чітку ізоляцію IoT-пристроїв від корпоративної мережі. Використання віртуалізації та контейнеризації з налаштуванням сегментації на рівні L2 OSI дозволило ефективно впровадити принцип мінімальних привілеїв. Реалізована топологія поєднує гнучке керування доступом через програмний маршрутизатор із гарантованою ізоляцією трафіку, що було підтверджено успішними практичними тестами та аналізом логів міжмережевого екрану.

За результатами тестування система продемонструвала повну відповідність вимогам: корпоративні клієнти мають доступ до зовнішніх ресурсів, тоді як прямий обмін даними з IoT-сегментом заблоковано. Подальші перспективи включають масштабування рішення до фізичної інфраструктури з використанням VLAN та інтеграцію систем виявлення вторгнень (IDS), що дозволить оптимізувати використання ресурсів та розширити функціонал захисту до рівня глибокого аналізу трафіку.