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

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-технічні засоби мережевого фаєрволу на основі одноплатної компютерної системи Raspberry Pi
Назва теми

КвРКІ 210379.21.04.25 ПЗ
Шифр

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

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

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

Виконав: студент IV курсу, група КІ2-21-4 _____ Денис ШЕВЧУК
Підпис Ініціали, прізвище

Керівник _____ Сергій ЛИСЕНКО
Підпис, дата Ініціали, прізвище

Нормоконтролер _____ Тетяна КИСІЛЬ
Підпис, дата Ініціали, прізвище

До захисту допускаю:
зав. кафедри комп'ютерної інженерії та інформаційних систем _____ Ольга ПАВЛОВА
Підпис Ініціали, прізвище

«05» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Освітній рівень бакалавр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

" 10 " 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Денису ШЕВЧУКУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно технічні засоби мережевого фаєрволу на основі одноплатної компютерної системи Raspberry Pi

Керівник проекту (роботи) Сергій ЛИСЕНКО, д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Програмно технічні засоби мережевого фаєрволу на основі одноплатної компютерної системи Raspberry Pi

Проєктування програмно технічного засобу мережевого фаєрволу на основі одноплатної компютерної системи Raspberry Pi

Програмно-апаратна реалізація програмно технічного засобу мережевого фаєрволу на основі одноплатної компютерної системи Raspberry Pi

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Архітектура ПЗ проєкту

Журнал тестування мережевого фаєрволу

Код програмного забезпечення

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КІІС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КІІС		

7. Дата видачі завдання «10» 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2025	виконано
4	Робота над розділом 2 – вибір компонентів для проєктування програмно технічного засобу мережевого фаєрволу на основі одноплатної компютерної системи Raspberry Pi	01.04.2025	виконано
5	Робота над розділом 3 – Програмно технічні засоби мережевого фаєрволу на основі одноплатної компютерної системи Raspberry Pi	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент _____ Денис ШЕВЧУК
Підпис Ініціали, прізвище

Керівник роботи _____ Сергій ЛИСЕНКО
Підпис Ініціали, прізвище

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1		КвРКІ 210379.21.04.25 ПЗ	Пояснювальна записка	57		
			Графічні матеріали			
2		КвРКІ 210379.21.04.25 Е8	Архітектура ПЗ проєкту	1		
3		КвРКІ 210379.21.04.25 Е8	Журнал тестування	1		
			мережевого фаєрволу			
4		КвРКІ 210379.21.04.25 Е8	Сценарій використання	1		
			фаєрволу			

Зм	Арк	№ докум	Підпис	Дата	КвРКІ 210379.21.04.25 ПЗ		
Розробив		Шевчук			Відомість проекту	Літера	Аркуш	Аркушів
Перевір.		Лисенко				У	1	1
Н. контр.		Кисіль		05.06.25		ХНУ, КІ2-21-4		
Затв.		Павлова						

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно технічні засоби мережевого фаєрволу на основі одноплатної компютерної системи Raspberry Pi».

Автор роботи: Денис ШЕВЧУК.

Керівник роботи: Лисенко Сергій Миколайович.

Пояснювальна записка: 55 с., 15 рис., 3 табл., 3 дод., 50 джерел.

Графічна частина: 3 креслення.

МЕРЕЖЕВИЙ ФАЄРВОЛ, RASPBERRY PI, VIRTUALBOX, IPTABLES, NAT, ФІЛЬТРАЦІЯ ТРАФІКУ

Метою роботи є розробка та програмно-апаратна реалізація мережевого фаєрволу на базі одноплатної комп'ютерної системи Raspberry Pi. Завдання полягає у створенні ефективного, гнучкого та масштабованого рішення для фільтрації мережевого трафіку, забезпечення безпеки локальної мережі та організації контролю доступу за допомогою інструментів Linux, зокрема iptables.

Об'єктом дослідження є програмно-апаратні засоби забезпечення мережевої безпеки.

Предметом дослідження є методи та засоби реалізації мережевого фаєрволу на базі одноплатної комп'ютерної системи Raspberry Pi.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.

Підпис студента

30.05.2025

Дата

ЗМІСТ

ВСТУП

У наш час стрімкого розвитку цифрових технологій інформація стала одним з найцінніших ресурсів як для окремих користувачів, так і для організацій у різних галузях. Передача, обробка та зберігання даних відбувається переважно через комп'ютерні мережі, що робить їх безпеку критично важливою складовою сучасної інфраструктури. В умовах глобальної доступності Інтернету та поширення мережевих сервісів, зокрема хмарних платформ, веб-додатків, IoT-пристроїв, зростає ймовірність атак з боку зловмисників. Це зумовлює необхідність надійного захисту мережевого периметру за допомогою спеціалізованих засобів – зокрема, фаєрволів.

Мережевий фаєрвол є одним з основних елементів системи інформаційної безпеки, завданням якого є контроль і фільтрація мережевого трафіку відповідно до встановлених правил. Він виконує функції виявлення і блокування потенційно небезпечних з'єднань, обмеження доступу до окремих мережевих ресурсів, а також формування базових політик безпеки на рівні мережевої взаємодії. Застосування фаєрволів дозволяє суттєво знизити ризик зовнішніх вторгнень, втрати конфіденційних даних та інших форм кіберзагроз.

Традиційно реалізація фаєрволів базувалася на використанні спеціалізованого обладнання з комерційним програмним забезпеченням, що потребувало значних фінансових витрат та технічних ресурсів. Однак із розвитком мікроелектроніки та програмного забезпечення з відкритим кодом з'явилась можливість створювати ефективні засоби захисту на базі малогабаритних та енергоефективних пристроїв. Одним з найпоширеніших прикладів таких пристроїв є одноплатний комп'ютер Raspberry Pi – універсальна платформа, яка при невисокій вартості забезпечує достатню обчислювальну потужність, широкі можливості підключення до мережі та високу гнучкість у конфігурації.

Використання Raspberry Pi як основи для побудови мережевого фаєрволу відкриває нові перспективи для впровадження бюджетних рішень у сфері

кібербезпеки, особливо в умовах обмежених ресурсів наприклад, у домашніх мережах, малих підприємствах, освітніх закладах або в межах навчального процесу. Платформа дозволяє встановити повноцінну операційну систему на базі Linux, у якій доступні інструменти фільтрації трафіку (iptables, nftables, Netfilter), моніторингу, журналювання подій, створення VPN-з'єднань та інших функцій, необхідних для побудови надійного фаєрволу.

У контексті зростаючої цифрової загрози питання розробки і впровадження доступних, ефективних та гнучких засобів захисту є надзвичайно актуальним. Використання Raspberry Pi у ролі фаєрволу дозволяє поєднати відкриту апаратно-програмну архітектуру, сучасні підходи до безпеки та практичну орієнтацію на реальні потреби користувачів. Це дає змогу створити систему, яка є не лише ефективною у технічному сенсі, а й доступною з погляду впровадження і обслуговування.

Дослідження, викладене у даній дипломній роботі, присвячене практичній реалізації програмно-технічного комплексу для забезпечення мережевого захисту з використанням Raspberry Pi. У межах роботи розглянуто особливості архітектури фаєрволу, обґрунтовано вибір операційного середовища, проведено налаштування фільтрації та маршрутизації трафіку, організовано журналювання та моніторинг, а також здійснено тестування ефективності створеного рішення. Застосований підхід дозволяє оцінити доцільність використання одноплатної комп'ютерної платформи для реалізації функцій мережевої безпеки та визначити її переваги й обмеження у порівнянні з класичними рішеннями.

1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ

1.1 Аналіз предметної області і виявлення наявних проблем і завдань

У сучасну епоху цифрових технологій, коли більшість процесів в освіті, бізнесі та побуті переходять в онлайн, безпека мережі стає важливою умовою стабільного функціонування інформаційних систем. Щодня зростає кількість кіберзагроз: від фішингу та вірусів до складних багаторівневих атак, які можуть вивести з ладу навіть великі компанії. Уразливими залишаються й звичайні користувачі, які часто не мають необхідного рівня захисту.

Для ілюстрації масштабу проблеми достатньо поглянути на статистику інцидентів кібербезпеки за останні роки (рис. 1.1) – крива атак постійно зростає, особливо в сегменті малих організацій та освітніх установ.

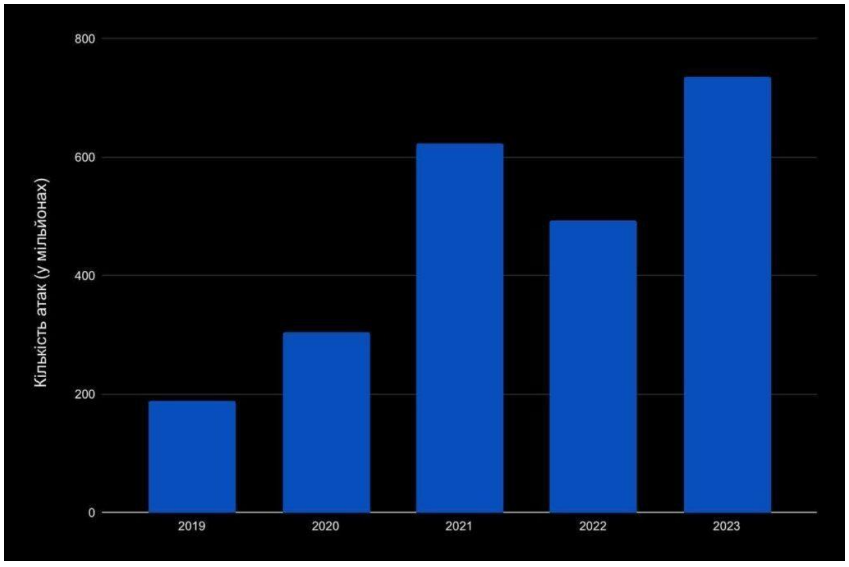


Рисунок 1.1 – Графік зростання кількості кіберзагроз [11]

Щоб протидіяти загрозам, широко використовуються фаєрволи – програмно-апаратні рішення, які здійснюють контроль і фільтрацію трафіку. Вони визначають, які з'єднання дозволяти або блокувати, допомагають уникнути сканування портів, шкідливих запитів і несанкціонованого доступу. Як правило, фаєрвол розташовується між зовнішньою мережею (Інтернетом) і внутрішньою інфраструктурою користувача, утворюючи першу лінію оборони.

Цю логіку можна наочно відобразити за допомогою простої схеми взаємодії між користувачем, фаєрволом та Інтернетом(рис.1.2).

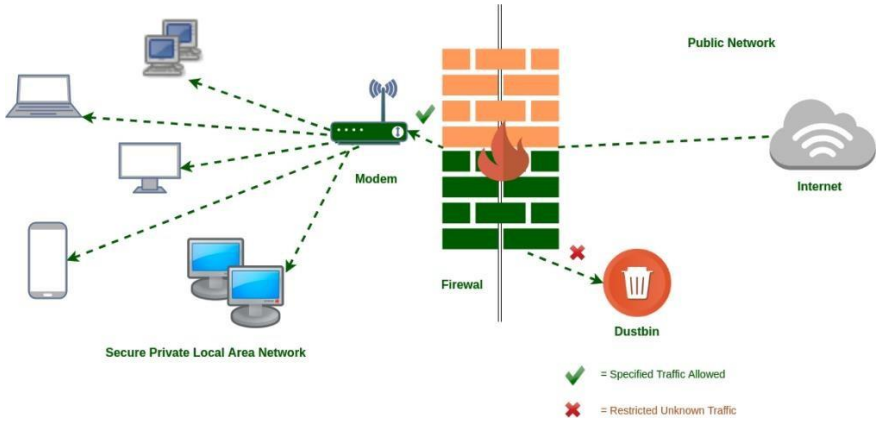


Рисунок 1.2 – Мережева схема – Інтернет → фаєрвол → внутрішня мережа [12]

Популярні рішення в цій галузі включають продукти від таких виробників, як Cisco, Fortinet, MikroTik, а також open-source системи – pfSense, OPNsense, Sophos XG. Вони надають потужний функціонал і можливості масштабування, але часто виявляються недоступними для малих організацій через високу вартість або складність налаштування.

Фаєрвол є однією з базових складових архітектури інформаційної безпеки. Проте комерційні апаратні чи програмні рішення часто мають високу вартість або потребують глибоких знань для налаштування. Це створює потребу в недорогих, зрозумілих і водночас ефективних рішеннях, доступних навіть для організацій з обмеженим бюджетом.

В умовах обмеженого бюджету оптимальним виходом може стати використання одноплатного комп’ютера Raspberry Pi. Це компактний і доступний пристрій, який підтримує операційні системи на базі Linux, має Ethernet-інтерфейс, USB-порти та достатню потужність для обробки базового трафіку. Він ідеально підходить для створення простого фаєрволу з використанням таких інструментів, як iptables, nftables, UFW або Pi-hole.

Прикладом такого пристрою є Raspberry Pi 4 (рис.1.3), який можна легко інтегрувати в локальну мережу та налаштувати як фільтр трафіку.

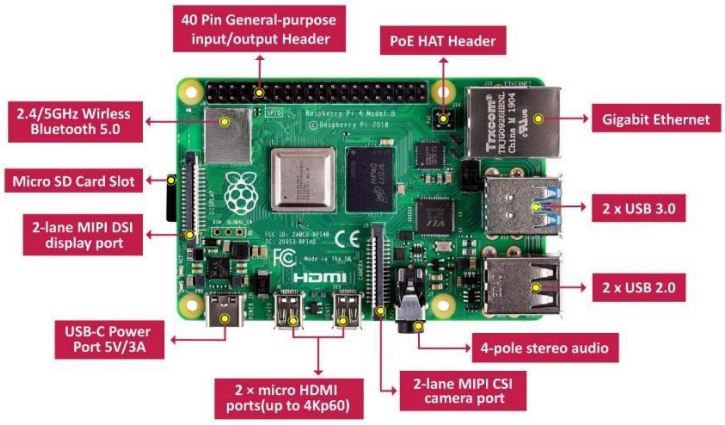


Рисунок 1.3 – Raspberry Pi з технічними характеристиками [13]

Проте слід враховувати і деякі обмеження. Raspberry Pi не розрахований на інтенсивне навантаження, а також вимагає знань у сфері адміністрування Linux. Для спрощення роботи можливе використання веб-інтерфейсів адміністрування, таких як Webmin або Cockpit, які дають змогу керувати фаєрволом без глибокого занурення в термінал.

Отже, ситуація на ринку IT-безпеки свідчить про наявність потреби у доступних і водночас ефективних рішеннях. Raspberry Pi у поєднанні з відкритими програмними засобами може стати відповіддю на це запитання – за умови правильної конфігурації, продуманої архітектури та зручного для користувача інтерфейсу.

1.2 Порівняльний аналіз переваг та недоліків існуючих рішень

На сучасному ринку існує велика кількість рішень для побудови мережевих фаєрволів, як апаратних, так і програмних. Вибір найбільш підходящого рішення залежить від багатьох факторів, зокрема, функціональних вимог, бюджету та рівня необхідного захисту. Кожна з них має свої сильні та слабкі сторони(табл.1.1)

Таблиця 1.1. Переваги та недоліки

Рішення	Переваги	Недоліки
Cisco ASA	Надійність, гнучкість, підтримка VPN, корпоративна підтримка	Висока вартість, складність налаштування
Fortinet	Розширений функціонал, інтеграція з іншими рішеннями	Дорогі ліцензії, складний інтерфейс
MikroTik	Гнучкість, поужні можливості скриптів, доступність	Крива навчання, незавжди зручний веб-інтерфейс
pfSense / OPNsense	Безкоштовні, гнучкі, модульна структура	Потребуюьб окремого ПК або мінікомп'ютера, технічні знання
Pi-hole + iptables/nftables (на Raspberry Pi)	Простота, дешевизна, відкритий код	Обмежена продуктивність, потреба в адмініструванні Linux

Cisco ASA (Adaptive Security Appliance) є одним із найвідоміших корпоративних рішень для побудови захищених мереж. Цей пристрій поєднує у собі функції класичного фаєрвола, системи виявлення та запобігання вторгнень (IDS/IPS), NAT, VPN, а також детального контролю доступу. Cisco ASA добре підходить для середніх і великих підприємств, оскільки забезпечує високу надійність, масштабованість і інтеграцію з іншими продуктами екосистеми Cisco. Водночас цей варіант вимагає спеціалізованих знань для налаштування й обслуговування та є досить дорогим у придбанні та підтримці.

Fortinet FortiGate – ще одне потужне апаратне рішення, яке часто використовується у корпоративному секторі. На відміну від традиційних фаєрволів, FortiGate пропонує функції фаєрвола нового покоління (Next Generation

Firewall), такі як глибока інспекція трафіку, фільтрація веб-контенту, антивірусний захист, VPN та контроль додатків. Fortinet орієнтований на побудову централізованих систем захисту, часто використовується у зв'язці з іншими продуктами компанії для створення єдиної безпекової інфраструктури. Однак, як і у випадку з Cisco, висока якість і функціональність супроводжуються значними витратами та потребою у високій кваліфікації персоналу.

MikroTik є популярним рішенням серед малого та середнього бізнесу, а також у домашніх мережах. Основу становить операційна система RouterOS, яка підтримує велику кількість функцій – фаєрвол, NAT, DHCP, VPN, фільтрація трафіку, управління смугою пропускання тощо. MikroTik відомий своєю доступністю, гнучкістю та широкими можливостями налаштування. Проте, інтерфейс системи може виявитися складним для новачків, особливо без досвіду роботи з мережею.

Серед програмних рішень варто згадати pfSense, який базується на FreeBSD і є повністю відкритим. pfSense встановлюється на звичайний комп'ютер або віртуальну машину та забезпечує повний набір функцій фаєрвола, VPN, балансування навантаження, моніторингу та аналітики трафіку. Його перевагою є гнучкість, постійна підтримка спільноти та можливість розширення за допомогою модулів. pfSense особливо популярний у навчальних закладах, серед IT-ентузіастів та у невеликих офісах. Недоліком може бути лише відсутність офіційної технічної підтримки у безкоштовній версії та потреба в базових знаннях роботи з UNIX-подібними системами.

Інші open-source альтернативи – OPNsense, Sophos XG (Home Edition), IPFire – також мають схожий функціонал і активно розвиваються. Вони орієнтовані на забезпечення максимальної безпеки за мінімальних витрат, що робить їх привабливими для невеликих організацій та користувачів з обмеженим бюджетом (рис 1.4).

Програмні фаєрволи, як pfSense, OPNsense або Sophos XG, є чудовою альтернативою, оскільки вони мають низьку вартість і надають велику кількість

функцій для керування безпекою мережі. Ці рішення можуть бути безкоштовними, але потребують певного досвіду для налаштування та адміністрування. Вони також дозволяють використовувати старе обладнання, що є важливою перевагою для організацій з обмеженим бюджетом. Проте, як і в випадку з апаратними рішеннями, програмні фаєрволи можуть бути проблемними, якщо потрібно обробляти великий обсяг мережевого трафіку.

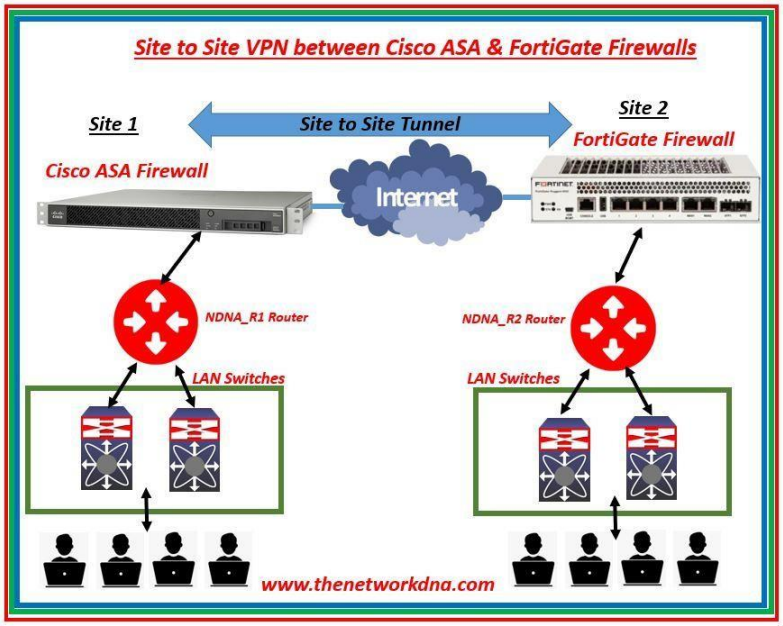


Рисунок 1.4 – Порівняння характеристик Cisco ASA [14]

У випадку використання Raspberry Pi як основи для фаєрволу, можна отримати доступне, економічне рішення. Raspberry Pi дозволяє створити ефективний фаєрвол на основі відкритого програмного забезпечення, такого як iptables або UFW. Це рішення має декілька переваг: низька вартість, енергозбереження та компактні розміри.

Однак Raspberry Pi має обмежену обчислювальну потужність, що може стати проблемою при великому обсязі трафіку. Також для налаштування та управління системою необхідні базові знання Linux та мережевих протоколів.

У контексті побудови мережевого фаєрволу для домашнього використання або малих офісів, вибір між Raspberry Pi, pfSense і MikroTik залежить від кількох

факторів, таких як вартість, вимоги до продуктивності та рівень гнучкості налаштувань.

Raspberry Pi, будучи одноплатним комп'ютером з обмеженими обчислювальними ресурсами, підходить для більш легких завдань, де не потрібна обробка великого обсягу мережевого трафіку.

Однак, завдяки відкритому коду і великій спільноті, Raspberry Pi надає можливість розширювати функціональність і налаштовувати систему за власним бажанням, використовуючи програмне забезпечення, таке як iptables, UFW або навіть Pi-hole для фільтрації реклами та шкідливих сайтів. Це чудовий варіант для тих, хто має базові знання Linux і шукає бюджетне рішення.

В свою чергу, pfSense є потужним інструментом з великими можливостями для налаштування та безпеки, включаючи підтримку VPN, IDS/IPS і різних методів фільтрації трафіку. Однак для його налаштування потрібні певні технічні знання, і для забезпечення стабільної роботи на високому рівні необхідне налаштування на більш потужному обладнанні, ніж Raspberry Pi.

Отже, кожне з існуючих рішень має свої переваги та недоліки. Апаратні фаєрволи підходять для великих організацій, які мають потребу в обробці великих обсягів трафіку та високому рівні безпеки. Програмні фаєрволи є кращим варіантом для малих і середніх організацій або для домашнього використання, оскільки вони дешевші і надають велику кількість можливостей для налаштування.

Raspberry Pi є економічно вигідним рішенням для тих, хто хоче створити бюджетний фаєрвол з мінімальними витратами, при цьому забезпечуючи достатній рівень захисту для невеликих мереж.

1.3 Підходи до вирішення задачі за темою дослідження

При розгляді підходів до вирішення задачі побудови фаєрволу на основі одноплатного комп'ютера Raspberry Pi, варто врахувати декілька ключових аспектів, серед яких гнучкість налаштувань, продуктивність і інтеграція з іншими

системами безпеки. Для цього можна використати різні методи та інструменти, зокрема, програмне забезпечення для фільтрації трафіку, побудову VPN-мереж, а також інтеграцію з системами IDS/IPS для виявлення та запобігання атак.

Один з основних підходів – використання стандартних інструментів фільтрації мережевого трафіку на базі Linux, таких як iptables або nftables. Ці інструменти дають змогу створювати ефективні правила для фільтрації пакетів, що проходять через фаєрвол, а також налаштовувати NAT (перенаправлення портів), що дозволяє організувати доступ до внутрішніх мереж або серверів. Одним із популярних рішень є використання UFW (Uncomplicated Firewall), що забезпечує зручніший інтерфейс для налаштування фаєрволу, дозволяючи швидко налаштовувати правила доступу, не вдаючись до складних команд у терміналі.

Ще один підхід полягає у використанні Pi-hole, який є спеціалізованим інструментом для блокування реклами та шкідливих сайтів на рівні мережевого шлюзу. Використовуючи Pi-hole разом з Raspberry Pi, можна не лише реалізувати базову функцію фаєрволу, але й додатково захистити домашню мережу від шкідливого контенту та зменшити навантаження на пристрої завдяки фільтрації реклами.

Для більш складних сценаріїв безпеки можна інтегрувати Snort або Suricata для реалізації функцій IDS/IPS. Ці системи аналізують мережевий трафік в реальному часі, шукаючи підозрілі або аномальні дії, що дозволяє виявляти потенційні загрози. Використання таких інструментів на Raspberry Pi дозволяє значно покращити рівень безпеки мережі, хоча варто враховувати, що це потребує додаткових ресурсів і може обмежити загальну продуктивність пристрою через обмежену обчислювальну потужність.

Ще одним підходом є створення VPN-сервера для забезпечення захищеного доступу до внутрішньої мережі. Використовуючи Raspberry Pi, можна налаштувати OpenVPN або WireGuard для створення приватної мережі, що дозволяє безпечно підключатися до мережі з віддалених точок. Це може бути особливо корисним для організацій або домогосподарств, де необхідно захистити передачу даних при

підключенні до загальнодоступних мереж.

Для полегшення адміністрування та забезпечення більш зручного інтерфейсу можна використовувати Webmin або Cockpit – графічні інтерфейси для керування сервером, що дозволяють налаштовувати фаєрвол без необхідності працювати через командний рядок. Ці інструменти спрощують роботу з мережею і дозволяють виконувати основні завдання, такі як управління користувачами, налаштування мережі та моніторинг трафіку, через веб-інтерфейс.

Загалом, підхід до вирішення задачі залежить від рівня складності, вимог до продуктивності і бюджету. Raspberry Pi, як дешевий та енергоефективний пристрій, може стати чудовою основою для побудови фаєрволу в умовах обмежених ресурсів, але для досягнення високого рівня безпеки та продуктивності буде необхідно виважено комбінувати різні інструменти і налаштування.

З програмної точки зору, у якості основного інструменту для реалізації політик фільтрації обрано iptables – потужний засіб керування трафіком у системах Linux. Він дає змогу створювати правила для обробки вхідного, вихідного і транзитного трафіку, враховуючи протоколи, порти, IP-адреси та стани з'єднання. Для полегшення конфігурації можливе використання допоміжних утиліт, таких як UFW або nftables (у випадку новіших систем).

Окремим напрямом є проектування архітектури фаєрвола, яка передбачає логічний поділ на зони безпеки (наприклад, зовнішня, внутрішня та демілітаризована зона – DMZ), а також визначення політик доступу між ними. У контексті Raspberry Pi реалізація такої моделі можлива за допомогою VLAN або кількох мережевих інтерфейсів (через USB-адаптери).

Також варто враховувати сучасні методи захисту: блокування небажаного DNS-трафіку, автоматичне оновлення чорних списків, використання SPI (stateful packet inspection), ведення журналів, а за потреби – налаштування VPN для безпечного віддаленого доступу до мережі.

У сукупності, запропонований підхід дозволяє створити доступний, функціональний і адаптований до реальних потреб фаєрвол, заснований на

відкритих технологіях. Це відповідає головній меті дослідження – пошуку практичного рішення для покращення кіберзахисту з мінімальними витратами.

1.4 Постановка задачі

Завдання полягає в розробці мережевого фаєрволу на основі одноплатного комп'ютера Raspberry Pi, який буде забезпечувати ефективний захист локальної мережі від зовнішніх загроз. Ціль – створити доступне та зручне рішення, яке дозволить користувачам з обмеженими технічними знаннями налаштовувати фаєрвол і забезпечувати основні функції безпеки.

Основними завданнями є фільтрація мережевого трафіку за допомогою таких інструментів, як iptables, nftables або UFW, а також налаштування VPN для забезпечення захищеного віддаленого доступу до мережі. Для цього необхідно вибрати оптимальне програмне забезпечення, яке буде легко інтегруватися з Raspberry Pi та працювати з обмеженими ресурсами.

Важливим елементом є спрощене адміністрування системи. Для цього потрібно створити або інтегрувати інтерфейс, який дозволить користувачам легко керувати налаштуваннями фаєрволу, не вдаючись до складних командних рядків. Це може бути графічний інтерфейс на основі Webmin або Cockpit, який зробить управління фаєрволом доступним навіть для новачків.

Завдання також передбачає інтеграцію з системами IDS/IPS, щоб виявляти і блокувати підозрілий трафік, що дозволить підвищити рівень безпеки мережі.

1.5 Висновки до першого розділу

У ході аналізу теоретичних основ дослідження було встановлено, що мережеві фаєрволи є критично важливими засобами забезпечення інформаційної безпеки в сучасному цифровому середовищі. З розвитком інтернет-технологій та зростанням кількості кіберзагроз виникає необхідність у доступних, ефективних і масштабованих засобах захисту як для великих організацій, так і для малих

підприємств чи домашніх користувачів.

На прикладі огляду апаратних і програмних рішень було виявлено, що промислові продукти, такі як Cisco ASA або Fortinet FortiGate, забезпечують високий рівень безпеки та функціональності, однак потребують значних фінансових вкладень і спеціалізованих знань для налаштування. Програмні рішення на кшталт pfSense, OPNsense або Sophos XG пропонують широкий спектр можливостей за менші витрати, але також вимагають базової технічної підготовки користувача. Водночас open-source рішення з використанням одноплатного комп'ютера Raspberry Pi стали перспективним напрямом для побудови бюджетних фаєрволів.

Порівняльний аналіз показав, що Raspberry Pi є доступною платформою, яка дозволяє реалізувати функціональний мережевий фаєрвол для невеликих мереж, зокрема з використанням інструментів, таких як iptables, nftables, UFW, Pi-hole, а також систем виявлення загроз Snort або Suricata. Raspberry Pi добре підходить для побудови фаєрволу у малих офісах або домашніх умовах, забезпечуючи базовий рівень захисту при мінімальних витратах. Обмеженнями такого підходу є невисока обчислювальна потужність пристрою та потреба в початкових знаннях адміністрування систем на базі Linux.

У розділі також розглянуто підходи до реалізації функціоналу фаєрволу на Raspberry Pi, які включають не лише фільтрацію трафіку, але й інтеграцію з VPN-сервісами, системами виявлення атак, веб-інтерфейсами адміністрування.

Raspberry Pi у поєднанні з відповідним програмним забезпеченням є перспективною платформою для розробки доступних і ефективних фаєрволів, які можуть бути адаптовані до вимог конкретного середовища. Дослідження підтвердило наявність чіткої потреби в таких рішеннях, особливо у сферах з обмеженим фінансуванням – в освітніх установах, невеликих компаніях та для персонального використання. Надалі доцільно сфокусуватися на практичній реалізації такої системи, враховуючи оптимальне поєднання простоти налаштування, продуктивності та рівня безпеки.

2 ПРОЄКТУВАННЯ ПРОГРАМНО ТЕХНІЧНОГО ЗАСОБУ МЕРЕЖЕВОГО ФАЄРВОЛУ НА ОСНОВІ ОДНОПЛАТНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ RASPBERRY PI

2.1 Визначення апаратних і програмних підсистем програмно-технічного засобу

Для реалізації та тестування програмно-технічного засобу мережевого фаєрволу було вирішено використати одноплатну комп'ютерну систему Raspberry Pi**.** Проте, з огляду на етап прототипування, а також для зручності експериментів і налагодження конфігурації, було обрано варіант емуляції Raspberry Pi в середовищі VirtualBox**.** Такий підхід дозволяє уникнути ризиків, пов'язаних з пошкодженням фізичного пристрою, спрощує процедуру резервного копіювання та пришвидшує процес тестування.

Raspberry Pi – це недорогий, енергоефективний одноплатний комп'ютер, що підтримує мережеві інтерфейси, зокрема Ethernet та Wi-Fi, і має змогу працювати під управлінням повноцінних Linux-дистрибутивів. Це робить його придатним для побудови фаєрволів, маршрутизаторів, точок доступу, серверів VPN тощо.

Основними перевагами Raspberry Pi як основи для фаєрволу є:

– підтримка мережевих протоколів і Linux-утиліт для фільтрації трафіку (iptables, nftables, firewalld);

– низька вартість та мала споживана потужність;

– наявність активної спільноти та великої кількості інструкцій з налаштування.

З програмної точки зору існувала можливість встановлення системи безпосередньо на фізичне обладнання або використання віртуалізації. Емулятори типу QEMU чи професійні середовища на кшталт VMware Workstation дають змогу імітувати роботу мережевого пристрою на будь-якій архітектурі. Проте, у цьому проєкті було обрано VirtualBox – зручну, безкоштовну та кросплатформну платформу віртуалізації.

Вибір VirtualBox обумовлений його широкими можливостями: підтримка знімків стану системи (snapshots), просте налаштування мережевих інтерфейсів, сумісність з різними дистрибутивами Linux і можливість запуску декількох віртуальних машин одночасно. Завдяки цьому стало можливим створити повноцінне віртуальне середовище, в якому фаєрвол моделюється без потреби у додатковому фізичному обладнанні. Це значно пришвидшило розробку, полегшило відлагодження правил фільтрації та забезпечило гнучкість у тестуванні різних конфігурацій без ризику для реальної мережі.

Для розробки була обрана віртуальна машина VirtualBox, яка дозволяє запускати Raspberry Pi OS як звичайну гостьову систему на хост-комп'ютері. Емуляція не дає повної апаратної сумісності, але є цілком достатньою для розробки конфігурацій фаєрволу, тестування сценаріїв маршрутизації та написання правил фільтрації. VirtualBox дозволяє створювати і керувати віртуальними машинами (ВМ), у яких можна запускати різноманітні операційні системи без потреби змінювати конфігурацію основної (host) системи. Це робить VirtualBox надзвичайно зручним інструментом для розробників, системних адміністраторів, а також у навчальному процесі.

У контексті проєктування фаєрволу на базі Raspberry Pi, VirtualBox став ключовим елементом на етапі розробки та моделювання. Основними перевагами використання цього середовища є:

- швидкість розгортання;
- масштабованість;
- безпека тестування;
- збереження станів;
- гнучкість налаштування мережі;
- підтримка спільних папок.

VirtualBox також дозволяє експортувати віртуальні машини в формати OVA або VDI, що дає змогу переносити готову систему на інший комп'ютер або запускати її у хмарному середовищі. Після завершення етапу розробки готову

конфігурацію легко перенести на фізичну Raspberry Pi, адаптувавши лише специфічні параметри системи.

Таким чином, VirtualBox виступив як ефективний інструмент віртуалізації для створення, тестування і вдосконалення фаєрволу без потреби у фізичному обладнанні на першому етапі.

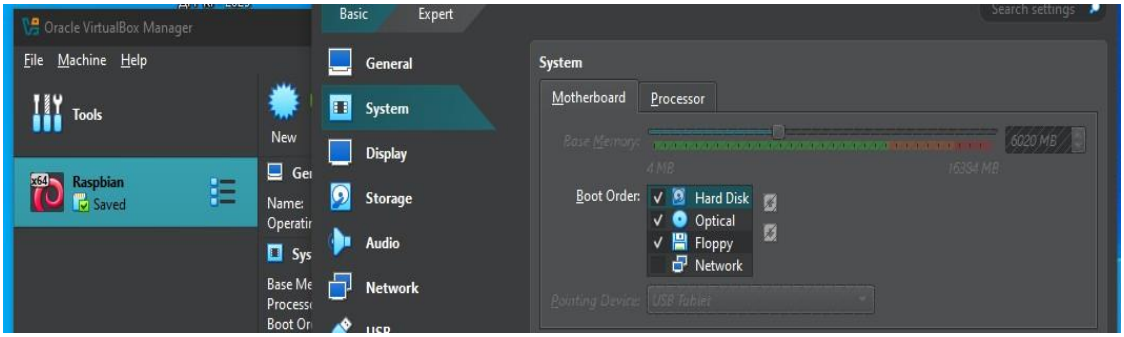


Рисунок 2.1 – Інтерфейс налаштування віртуальної машини

У рамках цього проєкту було створено віртуальну машину з параметрами, наближеними до характеристик реального Raspberry Pi: обсяг оперативної пам’яті 6 ГБ, динамічний віртуальний жорсткий диск на 4 ГБ, тип гостьової операційної системи – Debian 32-bit (рис.2.1). Як дистрибутив було обрано Raspberry Pi OS Bullseye (i386), оскільки ця система підтримує інтерфейсну та мережеву архітектуру, аналогічну до ARM-версій, але сумісну з архітектурою x86, що використовується VirtualBox.

Особливу увагу було приділено конфігурації мережевих інтерфейсів віртуальної машини. У ній були налаштовані два мережеві адаптери: перший – у режимі NAT для доступу до зовнішньої мережі (Інтернет), другий – у режимі внутрішньої мережі для моделювання локального середовища (рис 2.2). Така схема дозволила побудувати повноцінний фаєрвол, що фільтрує трафік між внутрішнім і зовнішнім сегментами, і тестувати його роботу в умовах, близьких до реальних.

VirtualBox надав можливість створювати знімки системи, що значно полегшило процес налагодження та тестування. У разі помилок конфігурації система легко поверталась до попереднього стабільного стану.

Також середовище дозволило швидко повторювати типові експерименти з мінімальними витратами часу. Завдяки цьому етап початкового проєктування пройшов швидко й ефективно.

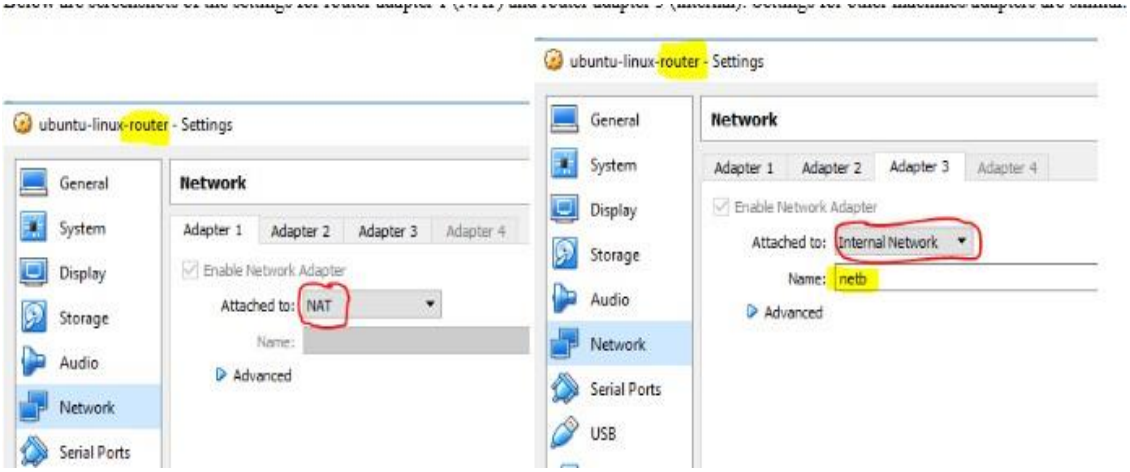


Рисунок 2.2 – Налаштування мережевих адаптерів[15]

Вибір саме Raspberry Pi OS пояснюється її повною сумісністю з інструментами, необхідними для реалізації фаєрволу: iptables, nftables, netfilter, а також підтримкою служб DHCP, DNS і NAT. Усі компоненти мають відкритий вихідний код і гнучкі можливості конфігурації, що є критично важливим для побудови надійного й адаптивного засобу фільтрації трафіку.

Таким чином, поєднання віртуального середовища VirtualBox з Raspberry Pi OS дозволило створити ефективну платформу для розробки фаєрволу, яка в подальшому може бути з легкістю перенесена на фізичне обладнання з мінімальними змінами в налаштуваннях. Це забезпечує гнучкість у розгортанні й масштабуванні рішення, а також дозволяє ефективно розмежувати етапи розробки та експлуатації.

2.2 Розробка архітектури системи мережевого фільтрування на базі Raspberry Pi

Розробка архітектури системи мережевого фільтрування на базі

одноплатного комп'ютера Raspberry Pi передбачає створення надійного, ефективного та водночас економічного рішення, здатного забезпечити безпеку локальної мережі від зовнішніх загроз. Raspberry Pi, завдяки своїм компактним розмірам, низькому енергоспоживанню та достатній обчислювальній потужності, є ідеальною платформою для побудови такого пристрою, який може виконувати функції міжмережевого екрану (фаєрволу) та маршрутизатора. В архітектурі системи передбачено наявність двох основних мережевих інтерфейсів. Перший інтерфейс, як правило, використовується для підключення до зовнішньої мережі, якою зазвичай є Інтернет.

Це може бути як дротове підключення через Ethernet порт (eth0), так і бездротове через інтерфейс wlan0. Другий мережевий інтерфейс призначений для підключення до локальної мережі користувача, і для цього часто застосовують додатковий USB-ethernet адаптер (eth1). Така двоінтерфейсна конфігурація дозволяє Raspberry Pi працювати як шлюз, через який проходить весь мережевий трафік, що дає можливість ефективно контролювати і фільтрувати пакети, що надходять і виходять (рис 2.3).

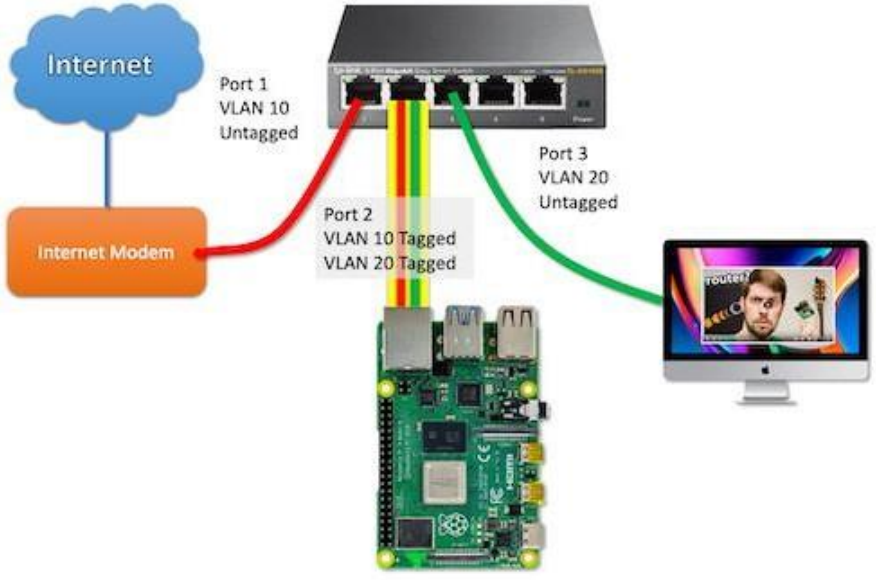


Рисунок 2.3 – Схема архітектури мережевого фільтрування[16]

Операційною системою для Raspberry Pi обирають Raspberry Pi OS, яка

базується на Debian Linux і забезпечує стабільну роботу з широким набором інструментів для управління мережею. Важливою складовою програмного забезпечення є пакет iptables або сучасніший nftables, які служать основою для створення та підтримки правил фільтрації трафіку. Ці інструменти дозволяють задавати складні умови, що регламентують, які пакети пропускати, а які блокувати, керуючись параметрами, такими як IP-адреси, порти, протоколи і стан з'єднання.

Для кращого розуміння, наведемо приклад типових правил фільтрації, які можна реалізувати за допомогою iptables(табл. 2.1)

Таблиця 2.1. Приклади правил фільтрації (iptables)

Правило	Опис	Приклад команди
Блокування вхідного трафіку на порт 80	Заборона HTTP-запити ззовні	Iptables -A INPUT -p tcp –dport 80 -j DROP
Дозвіл SSH доступу	Дозволити підключення через SSH з певної IP	Iptables -A INPUT -p tcp -s 192.168.1.100 –dport 22 -j ACCEPT
Заборона пінгів (ICMP)	Відхилити ICMP Echo Request (ping)	Iptables -A INPUT -p icmp –icmp-type echo-request -j DROP

IPTables – це міжмережевий екран (фаєрвол). Ми розглянемо, як проходить обробка пакетів через різні таблиці та ланцюжки в кожній з них. Ці знання будуть дуже корисними пізніше, коли ми почнемо створювати власні набори правил, особливо якщо в них будуть використовуватися такі дії, як DNAT, SNAT і, звичайно, TOS.

У цьому списку допускається виконувати тільки перелічені нижче дії:

- TOS;
- TTL;
- MARK.

Дія TOS відповідає за встановлення значень у полі Type of Service в IP-пакеті. Це поле використовується для визначення пріоритетів обробки пакету в мережі, тобто вказує, як саме пакет має маршрутизуватися. Втім, варто пам’ятати, що більшість інтернет-маршрутизаторів насправді ігнорує це поле. Тому не рекомендується змінювати значення TOS для пакетів, які прямують у глобальну мережу, адже там, де це поле враховують, можуть виникнути помилкові рішення щодо маршруту.

Дія TTL задає значення поля TTL (Time To Live) у пакеті. Це може бути корисно, якщо хочемо сховати наш фаєрвол від пильних інтернет-провайдерів. Деякі провайдери не люблять, коли до одного інтернет-з’єднання підключено кілька пристроїв. Вони дивляться на значення TTL у пакетах, щоб зрозуміти, чи використовується підключення одним комп’ютером чи кількома. Таким чином, зміна TTL допомагає уникнути такого контролю.

Дія MARK призначена для присвоєння пакету особливої позначки, яку потім можна використовувати в інших правилах iptables або в інших інструментах, наприклад, iproute2. Завдяки таким «міткам» можна контролювати маршрутизацію пакетів, обмежувати трафік та виконувати інші подібні дії.

Залежно від типу протоколу, пакети в ядрі можуть перебувати у різних станах. Водночас поза ядром існує лише чотири основні стани пакетів. Найчастіше для їх визначення застосовується параметр state, який може приймати такі значення: NEW, ESTABLISHED, RELATED та INVALID. Детальніше про кожен із цих станів можна дізнатися з таблиці, наведеної нижче.

Ці чотири стани використовуються у критерії state для фільтрації пакетів. Механізм визначення стану дозволяє значно підвищити ефективність і безпеку захисту. Раніше, щоб пропускати зворотній трафік у локальну мережу, доводилося відкривати всі порти понад 1024. Завдяки визначенню стану тепер це не потрібно – можна дозволяти лише відповідні відповіді на запити, при цьому блокуючи будь-які спроби зовнішніх підключень.

Почнемо з розгляду TCP-з'єднань. У цьому та наступних розділах ми

детально розглянемо, як визначаються стани та обробляються пакети для трьох основних протоколів – TCP, UDP і ICMP. Також торкнемося ситуацій, коли протокол не належить до цих трьох. Розпочнемо з TCP, адже саме він має низку особливостей у способі відстеження стану з'єднання в iptables.

TCP-з'єднання встановлюється за допомогою обміну трьома пакетами, які ініціюють і підтверджують з'єднання, по якому далі відбуватиметься передача даних. Спершу надсилається SYN-пакет, на який відповідає SYN/ACK, а потім отримується підтверджуючий ACK. Після цього з'єднання вважається активним і готовим до обміну інформацією. Можливо, ви запитаєте: «Як відстежується це з'єднання?». Насправді це працює досить просто.

Процес відстеження з'єднань для усіх типів майже ідентичний. На зображенні нижче ви можете побачити всі ключові етапи встановлення з'єднання. З боку користувача трасувальник не відстежує докладно весь процес – він просто, отримавши перший SYN-пакет, позначає його як NEW. Коли ж через нього проходить другий пакет – SYN/ACK, – з'єднання отримує статус ESTABLISHED. Чому це відбувається саме на другому пакеті? Про це трохи пізніше.

При налаштуванні правил ви можете дозволити вихідні пакети зі станами NEW і ESTABLISHED, а вхідні – лише з ESTABLISHED, і це буде працювати коректно. Якщо ж трасувальник постійно позначав би всі з'єднання як NEW, тоді встановити зв'язок із зовнішнім світом було б неможливо або довелося б відкривати вхідний трафік для пакетів NEW.

Для ядра ситуація складніша, оскільки там TCP-з'єднання проходять через кілька проміжних станів, які не відображаються на рівні користувача.

Окрім базового фаєрволу, для більшої функціональності та безпеки до системи можна додати такі сервіси, як dnsmasq для надання локального DNS і DHCP, fail2ban для захисту від повторних спроб несанкціонованого доступу (рис 2.4), а також утиліти tcpdump чи wireshark для глибокого аналізу мережевого трафіку.

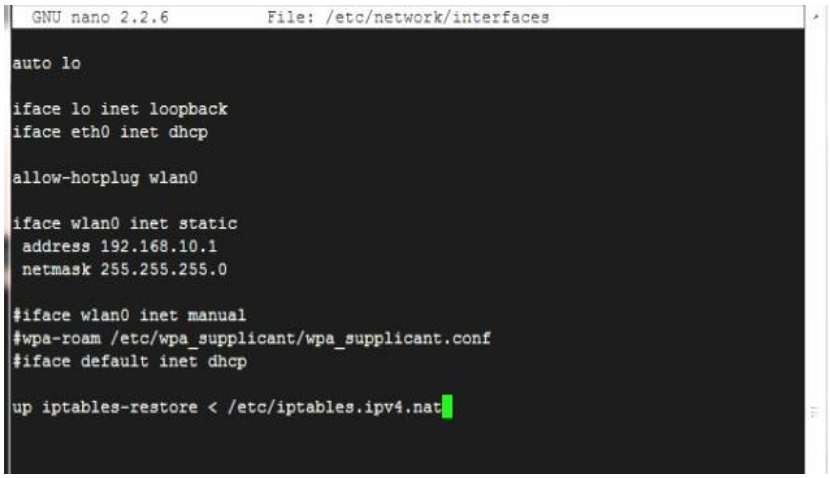


Рисунок 2.4 – Інтерфейс командного рядка[17]

Узагальнено, архітектура системи базується на тому, що весь трафік, який приходить з Інтернету, спочатку проходить через фаєрвол Raspberry Pi, де він аналізується за заздалегідь визначеними правилами. Якщо пакет відповідає вимогам безпеки та не порушує встановлених політик, він дозволяється до передачі у внутрішню мережу. І навпаки, підозрілі чи небажані пакети відхиляються або заносяться до журналів для подальшого аналізу. Зі сторони локальної мережі, пристрої також здійснюють вихідний доступ у зовнішні мережі через цей же шлюз, що дозволяє реалізувати контроль за вихідним трафіком і запобігати витоку конфіденційної інформації чи несанкціонованому доступу до зовнішніх ресурсів.

Окрім основних функцій фільтрації, система підтримує реалізацію трансляції мережевих адрес (NAT), що дозволяє локальним пристроям використовувати одну зовнішню IP-адресу для виходу в Інтернет. Це важливий аспект, який забезпечує приховування структури внутрішньої мережі від зовнішніх спостерігачів і підвищує безпеку загалом. Крім того, завдяки використанню операційної системи на базі Linux, систему можна легко масштабувати і розширювати: наприклад, додаючи підтримку VPN для захищеного віддаленого доступу, інтегруючи системи виявлення вторгнень (IDS/IPS) або централізоване логування подій.

Перевагою такої архітектури є її простота та доступність. Raspberry Pi – недорогий пристрій, який при цьому володіє достатнім функціоналом для побудови ефективного фаєрволу. Архітектура передбачає можливість гнучкого

налаштування, що дозволяє адаптувати правила фільтрації під конкретні потреби мережі – від найпростіших домашніх мереж до невеликих офісних середовищ. Також вона забезпечує незалежність від виробника і платформну гнучкість, оскільки заснована на відкритому програмному забезпеченні.

Візуально архітектуру можна уявити як систему, де Raspberry Pi розташований між зовнішньою мережею (Інтернетом) і внутрішньою локальною мережею, і виступає посередником, який ретельно контролює всі дані, що проходять крізь нього. Завдяки такій архітектурі можна бути впевненим, що внутрішня мережа захищена від більшості класичних атак, а адміністратор отримує можливість у будь-який момент змінювати налаштування, моніторити мережеву активність і швидко реагувати на загрози.

Таким чином, розроблена архітектура системи мережевого фільтрування на базі Raspberry Pi поєднує у собі простоту, доступність, гнучкість та надійність, що робить її оптимальним рішенням для захисту локальних мереж різного масштабу.

2.3 Конфігурування та створення правил фільтрації трафіку

Конфігурування мережевого фаєрволу на базі одноплатного комп’ютера Raspberry Pi є ключовим етапом реалізації системи безпеки локальної мережі. Цей процес включає не лише налаштування фізичних мережевих інтерфейсів, але й розробку та впровадження правил фільтрації мережевого трафіку, які визначають, який трафік дозволено пропускати, а який необхідно блокувати для захисту системи від потенційних загроз.

Першим кроком є конфігурування мережевих інтерфейсів. Raspberry Pi зазвичай має вбудований Ethernet-порт (eth0) та бездротовий адаптер (wlan0). Для побудови мережевого фаєрволу у конфігурації з двома інтерфейсами часто використовують додатковий USB-ethernet адаптер, який підключається як eth1. Це дозволяє розділити зовнішню мережу (Інтернет) та локальну мережу користувача. Для кожного інтерфейсу налаштовуються унікальні IP-адреси, маски підмережі,

шлюзи та параметри DNS. Такі налаштування можуть виконуватися вручну або за допомогою конфігураційних файлів і служб, наприклад dhcpcd.conf або netplan на Raspberry Pi OS.

Наступним важливим кроком є створення правил фільтрації трафіку. Для цього на Raspberry Pi використовують пакети iptables або сучасніший nftables, що є потужними інструментами для управління мережею в Linux. Ці інструменти дозволяють встановлювати правила, які керують проходженням мережевих пакетів через систему, аналізуючи їх за різними параметрами: IP-адресою відправника або одержувача, портами, протоколами (TCP, UDP, ICMP), станом з'єднання (нове, встановлене, відповіді тощо).

Правила в iptables організовані у ланцюги, які обробляють три основні типи трафіку: вхідний (INPUT), вихідний (OUTPUT) і транзитний (FORWARD). Вхідний трафік – це дані, що надходять ззовні на Raspberry Pi; вихідний – інформація, що передається від Raspberry Pi назовні; транзитний – трафік, що проходить через Raspberry Pi між зовнішньою і внутрішньою мережами. Грамотне налаштування кожного з цих ланцюгів дозволяє максимально контролювати мережеві зв'язки.

Для прикладу, у вхідному ланцюгу можна заборонити всі з'єднання, окрім дозволених, наприклад, доступу до SSH-сервера для адміністратора чи до веб-сервера. Аналогічно, у вихідному ланцюгу можна обмежити доступ внутрішніх пристроїв лише до певних інтернет-ресурсів, що суттєво знижує ризики витоку інформації або зараження шкідливим ПЗ. Всі небажані або підозрілі пакети можуть бути автоматично відхилені або логовані для подальшого аналізу(табл.2.2).

Таблиця 2.2 демонструє базовий набір правил для налаштування фаєрволу за допомогою утиліти iptables. Це мінімальний, але цілком функціональний сценарій фільтрації трафіку, який можна застосувати на одноплатному комп'ютері (наприклад, Raspberry Pi), що виконує роль маршрутизатора або шлюзу.

У наведеному прикладі використовуються основні ланцюги фільтрації ядра Linux: INPUT, OUTPUT, FORWARD, а також таблиця nat з ланцюгом

POSTROUTING, яка відповідає за трансляцію мережевих адрес (NAT).

Таблиця 2.2. Базові правила фільтрації трафіку

Ланцюг	Дія	Опис правил	Приклад команди
INPUT	ACCEPT	Дозволити SSH для адміністрування	iptables -A INPUT -p tcp –dport 22 -j ACCEPT
INPUT	ACCEPT	Дозволити HTTP (порт 80)	iptables -A INPUT -p tcp –dport 80 -j ACCEPT
INPUT	DROP	Відхилити всі інші вхідні з’єднання	iptables -P INPUT DROP
OUTPUT	ACCEPT	Дозволити весь вихідний трафік	iptables -P OUTPUT ACCEPT
FORWARD	ACCEPT	Дозволити транзитний трафік між інтерфейсами eth1 і eth0	iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
FORWARD	ACCEPT	Дозволити транзитний трафік у зворотному трафіку	iptables -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
POSTROUTING	MASQUERADE	Включити NAT для виходу локальних пристроїв в інтернет через eth0	iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

INPUT – відповідає за обробку вхідного трафіку, тобто трафіку, що надходить безпосередньо до самого пристрою (фаєрвола):

– перше правило дозволяє підключення по протоколу SSH (порт 22), що є необхідним для віддаленого адміністрування пристрою;

– друге правило дозволяє HTTP-з'єднання (порт 80), наприклад, якщо на пристрої запущено веб-сервер або інтерфейс керування;
– третє правило встановлює політику за замовчуванням на DROP, тобто всі інші вхідні підключення, які явно не дозволені, будуть блоковані. Це є хорошою практикою з безпеки – дозволяється лише необхідний трафік.

OUTPUT – визначає поведінку системи щодо вихідного трафіку, тобто трафіку, що генерується самим фаєрволом:

– у цьому прикладі дозволяється весь вихідний трафік, що підходить для більшості сценаріїв, коли сам пристрій може оновлюватися або звертатися до зовнішніх серверів (наприклад, для отримання DNS-запитів або пакетів оновлень).

FORWARD – контролює транзитний трафік, який проходить через пристрій між двома мережевими інтерфейсами:

– перше правило дозволяє трафік, що надходить з одного інтерфейсу (eth1) і перенаправляється на інший (eth0), що типово для локальної мережі, яка виходить в Інтернет;
– друге правило дозволяє зворотний трафік для вже встановлених або пов’язаних з’єднань, тобто відповіді на запити, ініційовані зсередини локальної мережі. Застосування модуля state (--state RELATED,ESTABLISHED) забезпечує контроль з’єднань на основі їхнього стану.

POSTROUTING (таблиця NAT) – використовується для трансляції вихідних IP-адрес локальних пристроїв на зовнішню адресу фаєрвола (інтерфейс eth0). Правило MASQUERADE динамічно підміняє адресу джерела на IP-адресу зовнішнього інтерфейсу. Це дає змогу всім пристроям у внутрішній мережі використовувати один зовнішній IP для виходу в Інтернет. Такий підхід є типовим для домашніх і малих корпоративних мереж.

Окрім фільтрації, у системі реалізується трансляція мережевих адрес (NAT), зокрема masquerading, що дозволяє локальним пристроям виходити в Інтернет

через одну спільну зовнішню IP-адресу. Це не лише оптимізує використання адресного простору, а й приховує структуру внутрішньої мережі від зовнішніх спостерігачів, що є додатковим засобом безпеки.

Резервне копіювання конфігураційних файлів є надзвичайно важливим аспектом управління будь-якою системою, включно з мережевим фаєрволом на базі Raspberry Pi. Конфігураційні файли містять усі налаштування, правила фільтрації трафіку, параметри мережевих інтерфейсів та інші критичні дані, необхідні для коректної роботи системи. Втрата або пошкодження цих файлів може призвести до збоїв у роботі фаєрволу, втрати доступу до мережі або навіть відкриття уразливостей.

Збереження копій конфігурацій дозволяє у разі помилки або некоректного налаштування швидко повернути систему до робочого стану, мінімізуючи час простою і ризики для безпеки. Наприклад, якщо після внесення змін у правила iptables фаєрвол перестає працювати належним чином, резервна копія дозволяє легко відкотитися до попередньої стабільної версії конфігурації. Це особливо важливо, коли система працює у виробничому середовищі, де навіть короткочасні збої можуть мати серйозні наслідки.

Оптимально зберігати резервні копії у кількох місцях: локально на самому Raspberry Pi у окремій директорії, а також на зовнішньому носії або віддаленому сервері. Автоматизація процесу резервного копіювання, наприклад, за допомогою скриптів або систем планування завдань (cron), дозволяє забезпечити регулярне оновлення копій без участі адміністратора, що підвищує надійність збереження даних.

Правила фаєрволу зазвичай описуються у вигляді набору команд, які зберігають у скриптах для автоматичного застосування під час запуску системи. Це гарантує, що при кожному перезавантаженні Raspberry Pi конфігурація мережевого фаєрволу буде відновлена у повному обсязі без додаткового ручного втручання. Для цього використовують init-скрипти, системні юніти systemd або спеціальні утиліти, які виконують команди iptables-restore чи nft -f.

Важливим аспектом є тестування правил фільтрації. Після застосування конфігурації необхідно перевірити, що дозволені служби коректно функціонують, а небажані з'єднання успішно блокуються. Для цього застосовують інструменти мережевого моніторингу, такі як tcpdump, wireshark, а також прості утиліти ping, traceroute або netcat для перевірки доступності портів.

Крім основних функцій фільтрації, конфігурація може бути доповнена сервісами, що підвищують рівень безпеки: наприклад, fail2ban автоматично блокує IP-адреси після кількох невдалих спроб входу, dnsmasq надає локальний DNS та DHCP сервер, що дозволяє краще контролювати мережеві ресурси.

Безпека самого Raspberry Pi є не менш важливою складовою загальної захищеності мережевого фаєрволу, адже навіть найкраще налаштований фаєрвол не зможе ефективно працювати, якщо сама платформа піддається атакам або має вразливості. Тому для забезпечення надійного захисту необхідно впровадити низку додаткових заходів безпеки, що допоможуть мінімізувати ризики компрометації системи.

Перш за все, важливо регулярно оновлювати операційну систему Raspberry Pi OS та всі встановлені пакети програмного забезпечення. Розробники системи та спільнота постійно випускають оновлення, які виправляють виявлені помилки, закривають відомі вразливості і покращують загальну стабільність роботи. Для оновлення використовують стандартні команди Linux, такі як sudo apt update і sudo apt upgrade. Регулярне оновлення забезпечує захист від більшості відомих атак і гарантує сумісність із новими версіями програм.

Другим важливим заходом є обмеження доступу до SSH – основного протоколу віддаленого адміністрування Raspberry Pi. За замовчуванням SSH часто налаштований на автентифікацію за паролем, що може бути вразливим до підбору або атаки перебором. Для підвищення безпеки рекомендується відмовитися від використання паролів і налаштувати автентифікацію за допомогою пари криптографічних ключів (публічного та приватного). Це значно ускладнює несанкціонований доступ, оскільки зловмиснику необхідно мати приватний ключ,

а не лише знати пароль. Для цього на клієнтському комп’ютері генерують пару ключів командою ssh-keygen, після чого публічний ключ копіюють на Raspberry Pi у файл ~/.ssh/authorized_keys. У конфігураційному файлі SSH сервера (/etc/ssh/sshd_config) рекомендується заборонити вхід за паролем (PasswordAuthentication no) та перезапустити службу.

Третім додатковим заходом безпеки є використання програми fail2ban. Ця утиліта автоматично відстежує спроби несанкціонованого доступу, наприклад, численні невдалі спроби входу через SSH, і блокує IP-адреси, з яких походять підозрілі дії, на певний період часу. Це значно знижує ймовірність успішної атаки методом перебору паролів (brute-force). fail2ban конфігурується через набори правил, які можна адаптувати під конкретні сервіси, а також забезпечує запис у логи для подальшого аналізу безпекових інцидентів.

Додатково рекомендується мінімізувати кількість відкритих сервісів на Raspberry Pi, видаляти непотрібні пакети та налаштовувати брандмауер таким чином, щоб лише необхідні порти були відкриті для зовнішнього доступу. Важливо також змінити стандартне ім’я користувача pi на інше, менш очевидне, щоб ускладнити пошук цілі атаки.

Підсумовуючи, комплексний підхід до безпеки Raspberry Pi, що включає регулярне оновлення ОС, захищений доступ через SSH з використанням ключів, а також захист від несанкціонованих спроб входу через fail2ban, дозволяє значно підвищити стійкість всієї системи мережевого фаєрволу. Такий підхід знижує ризики компрометації пристрою та забезпечує надійний захист мережі, що обслуговується цим пристроєм.

Таким чином, правильне і детальне конфігурування мережевого фаєрволу на Raspberry Pi, включно зі створенням гнучких та адаптивних правил фільтрації трафіку, є основою надійного захисту локальної мережі. Цей підхід дозволяє не тільки блокувати потенційні загрози, а й забезпечувати стабільність та безперервність роботи мережевих сервісів.

2.4 Висновки до другого розділу

У другому розділі було розглянуто основні аспекти проєктування програмно-технічного засобу мережевого фаєрволу на базі одноплатного комп'ютера Raspberry Pi. Проведено вибір апаратної платформи, яка завдяки своїм компактним розмірам, низькому енергоспоживанню та достатній обчислювальній потужності є оптимальним рішенням для створення ефективного та економічного фаєрволу. Визначено операційну систему Raspberry Pi OS як стабільне та гнучке середовище, що підтримує необхідні мережеві інструменти для конфігурування та управління фільтрацією трафіку.

Також детально описано архітектуру системи, яка передбачає використання двох мережевих інтерфейсів для розмежування зовнішнього та внутрішнього мережевого трафіку, що забезпечує ефективний контроль і фільтрацію пакетів. Було надано приклади конфігурації мережевих інтерфейсів та створення правил фільтрації за допомогою інструментів iptables, що дозволяє реалізувати як базові, так і більш складні політики безпеки.

Особливу увагу приділено заходам безпеки самої платформи Raspberry Pi, зокрема регулярним оновленням ОС, захищеному доступу через SSH із використанням ключів замість паролів та впровадженню утиліти fail2ban для захисту від атак перебору. Крім того, підкреслено важливість резервного копіювання конфігураційних файлів, що забезпечує швидке відновлення працездатності системи у разі збоїв або помилок.

Розроблена архітектура та підхід до налаштування програмно-технічного засобу мережевого фаєрволу на базі Raspberry Pi забезпечують надійний, гнучкий та масштабований захист локальної мережі, що робить це рішення ефективним для застосування у різних середовищах – від домашніх мереж до малих офісів.

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ МЕРЕЖЕВОГО ФАЄРВОЛУ НА ОСНОВІ ОДНОПЛАТНОЇ КОМПЮТЕРНОЇ СИСТЕМИ RASPBERRY PI

3.1 Опис реалізації модулів програмно-технічного засобу

Для створення мережевого фаєрволу обрана платформа Raspberry Pi OS i386, розгорнута у віртуальному середовищі VirtualBox. Віртуалізація забезпечує гнучкість у розробці, тестуванні та налагодженні системи без необхідності використовувати фізичний Raspberry Pi на початкових етапах. Такий підхід дозволяє швидко коригувати конфігурації, моделювати роботу мережі і забезпечувати безпечне середовище для експериментів.

Ідеальною апаратною платформою є одноплатний комп'ютер Raspberry Pi, який має достатню продуктивність для обробки мережевого трафіку в невеликих і середніх мережах. Наявність Ethernet-порту дозволяє підключати його як до локальної мережі, так і до Інтернету. Однак на початкових етапах через відсутність фізичного пристрою була обрана віртуалізація.

VirtualBox – це потужний інструмент, який дозволяє створювати віртуальні машини на будь-якій сучасній ОС. Він підтримує створення мережевих адаптерів у різних режимах: NAT, Bridged, Host-only. Для симуляції мережевого фаєрволу було налаштовано два мережеві інтерфейси – внутрішній (наприклад, host-only) та зовнішній (bridged або NAT), що відображають внутрішню локальну мережу і підключення до Інтернету відповідно.

Для розгортання середовища в VirtualBox було створено віртуальну машину з архітектурою x86 та встановлено Raspberry Pi OS (версія i386, Bullseye). У налаштуваннях ВМ було вказано дві мережеві карти: одна у режимі NAT (для доступу до інтернету), інша в режимі "внутрішньої мережі" або "bridge" (для емуляції внутрішньої локальної мережі) (рис.3.1). Це дозволило реалізувати базову схему маршрутизації та фільтрації трафіку між зовнішньою і внутрішньою мережами.

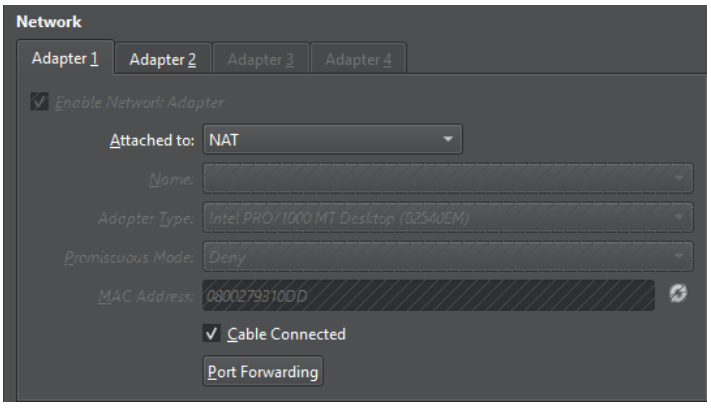


Рисунок 3.1 – Перша мережа NAT

У середовищі Linux найбільш поширеним способом реалізації NAT є використання модуля MASQUERADE у таблиці nat інструменту iptables. Цей модуль автоматично підставляє зовнішню IP-адресу в заголовки пакетів, що виходять у зовнішню мережу, і веде облік відповідностей для зворотного трафіку.

Таким чином, NAT – це невід'ємний компонент у створенні фаєрволу на базі Raspberry Pi, який дозволяє реалізувати функцію спільного доступу до інтернету та захищає внутрішню мережу від прямого доступу ззовні.

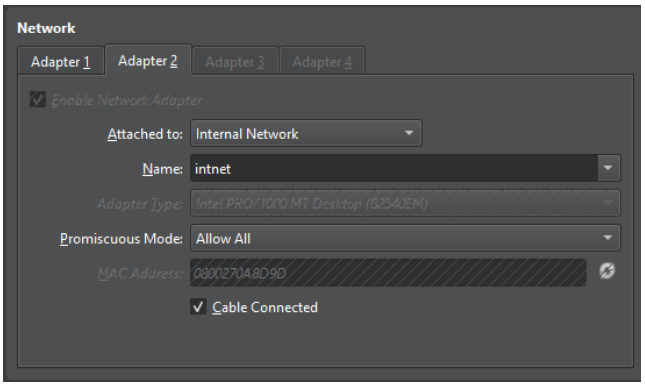


Рисунок 3.2 – Друга мережа Internet Network

Другий адаптер, налаштований у режимі доступу до зовнішньої мережі (Internet Network), виконував функцію вихідного інтерфейсу, через який віртуальна система отримувала доступ до глобального інтернету (рис.3.2). Його застосування дозволило розділити локальний трафік і зовнішній, що є типовим підходом для

реальних фаєрволів, які розташовуються між внутрішньою мережею та зовнішнім середовищем.

Цей адаптер дав змогу реалізувати трансляцію мережевих адрес (NAT), зокрема за допомогою механізму маскараду (MASQUERADE). Таким чином, пристрої внутрішньої мережі могли виходити в інтернет, використовуючи одну публічну IP-адресу. Це важливо для тестування правил iptables у наближених до реальних умовах. Другий мережевий адаптер фактично імітував зовнішній інтерфейс маршрутизатора, що дозволяло перевірити ефективність налаштувань маршрутизації, фільтрації та захисту трафіку у двох напрямках – ззовні та всередину системи.

Операційна система Raspberry Pi OS базується на Debian Linux, що забезпечує стабільність, сумісність з мережевими інструментами та великою спільнотою користувачів. Вибір 32-бітної версії i386 обумовлений необхідністю запуску ОС у VirtualBox, яка має архітектуру x86, на відміну від ARM в реальному Raspberry Pi.

Усередині ОС встановлено і налаштовано основні утиліти, необхідні для роботи фаєрволу: мережеві інструменти (ifconfig, ip), iptables для управління фільтрацією пакетів, а також додаткові пакети для моніторингу та налаштування.

iptables – це стандартний інструмент Linux для створення правил фільтрації та маршрутизації мережевого трафіку. У нашій системі він відповідає за контроль пакетів, що проходять через фаєрвол. Основні правила виглядають наступним чином:

- iptables -A INPUT -p tcp --dport 22 -j ACCEPT, дозволити вхідні SSH-з'єднання для адміністрування;
- iptables -A INPUT -p tcp --dport 80 -j ACCEPT, дозволити вхідний HTTP-трафік (порт 80);
- iptables -P INPUT DROP, відхилити всі інші вхідні пакети за замовчуванням;
- iptables -P OUTPUT ACCEPT, дозволити весь вихідний трафік;
- iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT, дозволити транзитний

трафік між інтерфейсами eth1 і eth0;

– iptables -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT, дозволити зворотній транзитний трафік (RELATED, ESTABLISHED);
– iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE, налаштувати NAT маскарадинг для виходу у Інтернет через eth0.

Ці правила забезпечують базовий рівень захисту і дозволяють лише легітимний трафік. Зокрема, фаєрвол приймає вхідні SSH-з’єднання для адміністрування, HTTP-запити, пропускає весь вихідний трафік і організовує NAT для локальних пристроїв, які виходять у Інтернет.

Завдяки віртуалізації у VirtualBox було змодельовано реальну роботу мережі з двома інтерфейсами, що полегшує налагодження. Перевірка правил iptables виконувалась за допомогою таких інструментів, як tcpdump і ping, що дозволяє відстежувати проходження пакетів і виявляти можливі помилки у конфігурації.

Окрім стандартних правил, була реалізована підтримка відстеження стану з'єднань (stateful firewall), що дозволяє пропускати лише пакети, які є частиною встановлених або пов’язаних з'єднань, що значно підвищує безпеку і ефективність фаєрволу.

Для перевірки роботи реалізованого фаєрволу була створена тестова модель мережі у віртуальному середовищі VirtualBox. Основною метою було перевірити, як фаєрвол обробляє вхідний, вихідний та транзитний трафік, а також оцінити ефективність реалізованих правил фільтрації.

Віртуальна мережа складалася з двох машин. Перша – віртуальна машина з Raspberry Pi OS, яка виступала як мережевий шлюз із встановленим фаєрволом на базі iptables. Ця машина мала два мережеві адаптери: один із типом "Internal Network" – для імітації локального інтерфейсу (eth1), а інший – з типом "NAT" або "Bridged Adapter", який забезпечував вихід у зовнішню мережу (eth0). Друга віртуальна машина, яка моделювала клієнтський пристрій внутрішньої мережі, була підключена до того ж внутрішнього адаптера, що й шлюз.

За такої конфігурації трафік від клієнта надходив на фаєрвол через внутрішній інтерфейс і надалі пересилався у зовнішню мережу або блокувався відповідно до заданих правил. Для перевірки правильності маршрутизації та фільтрації використовувались базові мережеві інструменти.

Команда ping дозволяла перевірити доступність зовнішніх ресурсів і відповідну обробку ICMP-пакетів. За допомогою curl тестувався HTTP-трафік на порт 80 – таким чином перевірялося, чи дозволено фаєрволом проходження веб-запитів. Утиліта netcat (nc) використовувалася для створення простих TCP/UDP-з'єднань, що дозволяло перевірити відкритість портів та реакцію фаєрволу на несанкціоновані спроби з'єднання.

Крім того, для аналізу роботи правил iptables було активовано логування. Правила з міткою LOG дозволяли зберігати у системному журналі повідомлення про пакети, які були заблоковані або дозволені, що значно спрощувало відлагодження конфігурації та дозволяло спостерігати за поведінкою трафіку в реальному часі.

Обраний підхід, що базується на використанні VirtualBox у поєднанні з Raspberry Pi OS i386 та інструментом iptables, дозволяє створити ефективне, гнучке і масштабоване рішення для мережевого захисту. Така реалізація забезпечує низку важливих переваг. Насамперед, немає потреби у фізичному Raspberry Pi, що дозволяє знизити витрати на апаратне забезпечення й оперативно розгорнути середовище розробки. Віртуалізація значно спрощує налаштування і дає змогу швидко змінювати конфігурацію, експериментувати з різними варіантами мережевої топології та миттєво перевіряти внесені зміни. Завдяки використанню стандартних Linux-інструментів з відкритим вихідним кодом забезпечується висока прозорість системи, гнучкість у налаштуванні правил фільтрації та загальна надійність функціонування. Крім того, така структура дає змогу легко інтегрувати додаткові засоби моніторингу або веб-інтерфейси для адміністрування, що підвищує зручність керування фаєрволом навіть для користувачів із базовим рівнем знань

3.2 Реалізація основного функціоналу системи

Було змодельовано просту локальну мережу, у якій віртуальна машина виконувала роль мережевого шлюзу та фаєрволу. Ця віртуальна машина мала два мережеві інтерфейси: eth0 і eth1. Інтерфейс eth0 був підключений до NAT-мережі VirtualBox, що дозволяло імітувати підключення до глобальної мережі Інтернет. Через цей інтерфейс проходив увесь зовнішній трафік, який надходив до віртуальної машини та виходив з неї.

Інтерфейс eth1 використовувався для зв’язку з внутрішньою локальною мережею, в якій розміщувалася клієнтська машина. Клієнтська машина через цей інтерфейс отримувала доступ до ресурсів, а також могла надсилати запити в Інтернет. Усі пакети, що йшли від клієнта в сторону Інтернету або навпаки, проходили через фаєрвол, де застосовувалися налаштовані правила фільтрації трафіку за допомогою iptables.

Ця структура дозволяла змоделювати реальне середовище роботи мережевого шлюзу, де фаєрвол контролює, які з'єднання дозволені, а які – заблоковані. За допомогою такої конфігурації було можливо тестувати різні сценарії, наприклад, пропускання дозволеного трафіку, блокування небажаних пакетів, а також роботу NAT (мережевого транслятора адрес) для забезпечення доступу внутрішніх пристроїв до Інтернету через один публічний IP-адрес.

Окрім того, використання VirtualBox дозволяло легко створювати, модифікувати і відтворювати таку мережеву модель без необхідності в фізичному обладнанні, що значно полегшувало розробку, тестування і налагодження мережевого фаєрволу на базі Raspberry Pi OS i386 з iptables. Цей підхід дозволив ефективно відстежувати і контролювати мережевий трафік, оцінювати роботу правил безпеки і забезпечувати ізоляцію між внутрішньою мережею і зовнішнім світом у зручному віртуальному середовищі.

Після успішного запуску віртуальної машини з операційною системою Raspberry Pi OS i386 наступним важливим кроком стало налаштування мережевих

інтерфейсів, які відіграють ключову роль у роботі мережевого фаєрволу. Для коректної роботи системи та забезпечення зв'язку між локальною мережею і Інтернетом необхідно було правильно сконфігурувати два основні інтерфейси: eth0 і eth1.

Спершу за допомогою команд ifconfig або ip a було проведено перевірку поточного стану мережевих інтерфейсів, їхніх налаштувань і призначених IP-адрес. Команда ifconfig виводить докладну інформацію про всі мережеві інтерфейси системи, включаючи їхні фізичні адреси (MAC), призначені IP-адреси, стан інтерфейсу (активний чи ні) та іншу корисну інформацію (рис.3.3). Аналогічно команда ip a (скорочено від ip address) надає більш сучасний і гнучкий спосіб перегляду та керування мережевими інтерфейсами в Linux-системах.

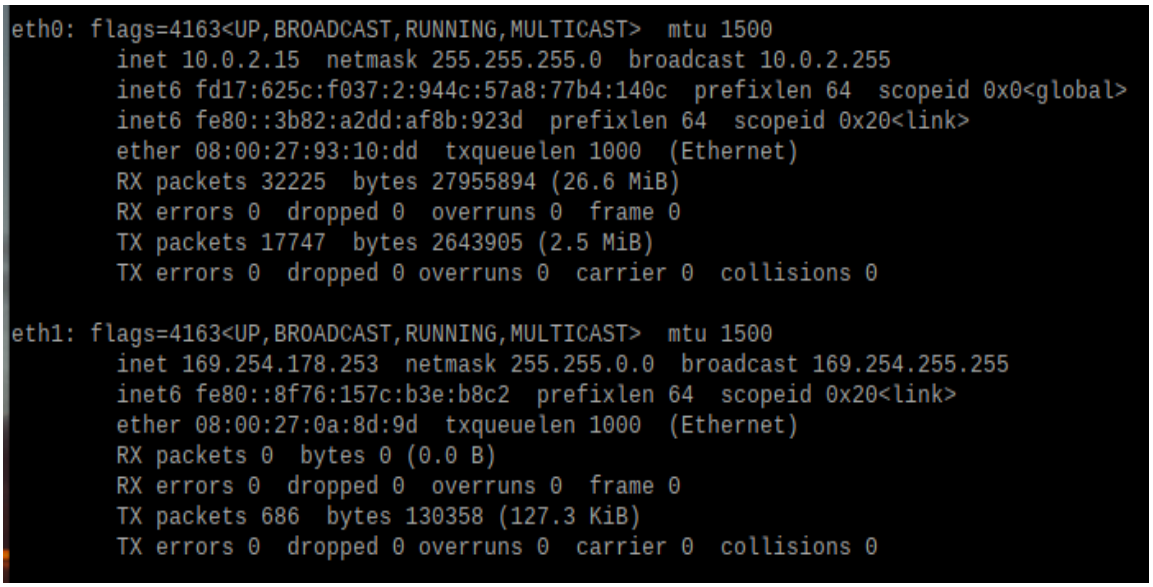


Рисунок 3.3 – Перевірка стану мережевих інтерфейсів

Інтерфейс eth0 був налаштований на отримання IP-адреси автоматично через механізм DHCP, оскільки він підключений до NAT-підмережі VirtualBox. Це означає, що при кожному запуску віртуальної машини eth0 звертається до віртуального DHCP-сервера VirtualBox і отримує динамічну IP-адресу, яка належить до підмережі NAT. Така адресація дозволяє забезпечити зв'язок із зовнішнім світом, тобто з Інтернетом, через NAT-мережу VirtualBox, яка виступає

посередником між віртуальною машиною і зовнішньою мережею.

З іншого боку, інтерфейс eth1 було налаштовано вручну, призначивши йому статичну IP-адресу. Це необхідно для організації стабільного і передбачуваного з'єднання з внутрішньою локальною мережею, де розташовується клієнтська машина. Статична адреса дозволяє постійно ідентифікувати шлюз у локальній мережі, що спрощує налаштування клієнтів і забезпечує надійність комунікації. Адресація eth1 здійснювалася в межах приватного діапазону IP, наприклад 192.168.1.1 з маскою підмережі 255.255.255.0, що відповідає класу C приватних мереж.

Такий поділ і налаштування інтерфейсів є типовою практикою у створенні мережевого шлюзу чи фаєрволу: один інтерфейс підключається до зовнішньої мережі з динамічною адресацією, а другий – до внутрішньої мережі з постійною статичною IP-адресою. Це забезпечує ефективне розділення трафіку, безпеку і можливість гнучкого керування правилами фільтрації пакетів.

Для налаштування статичної IP-адреси на eth1 використовувалися конфігураційні файли операційної системи, такі як /etc/dhcpcd.conf або /etc/network/interfaces (залежно від конкретної версії Raspberry Pi OS). У цих файлах було прописано фіксовану адресу, маску підмережі і, за потреби, шлюз та DNS-сервери для локальної мережі. Після внесення змін було виконано перезапуск мережевого сервісу або перезавантаження системи, щоб нові налаштування вступили в дію.

Крім того, перевірка працездатності інтерфейсів проводилася за допомогою команд ping і traceroute для підтвердження наявності зв'язку між клієнтом, фаєрволом і зовнішнім світом. Це допомагало переконатися, що IP-адресація і маршрутизація налаштовані коректно, а трафік проходить через потрібні інтерфейси відповідно до поставлених завдань.

Налаштування мережевих інтерфейсів eth0 і eth1 є фундаментальним етапом у побудові мережевого фаєрволу на основі Raspberry Pi OS у віртуальному середовищі VirtualBox. Правильна конфігурація інтерфейсів забезпечує ефективне

розділення внутрішнього і зовнішнього трафіку, що є необхідною умовою для подальшої побудови правил фільтрації, маршрутизації і забезпечення безпеки мережі.

Налаштування фаєрволу з використанням iptables є одним із ключових етапів реалізації мережевого захисту на основі Raspberry Pi OS. Інструмент iptables – це потужний і гнучкий засіб для керування мережевими пакетами в Linux, що дозволяє створювати складні правила фільтрації, контролювати доступ до мережевих сервісів, організовувати маршрутизацію і забезпечувати безпеку системи.

Основною метою налаштування було забезпечити захист внутрішньої мережі від несанкціонованого доступу, одночасно дозволивши необхідний трафік для нормального функціонування служб і забезпечення зв'язку з Інтернетом. Для цього було сформовано набір правил, які розподілені по основних ланцюгах iptables – INPUT, OUTPUT, FORWARD і NAT (POSTROUTING).

Правила у ланцюзі INPUT визначають, які пакети можуть потрапити до самого пристрою-фаєрволу, тобто до Raspberry Pi. У нашому випадку було важливо дозволити доступ до певних служб, які необхідні для адміністрування і роботи системи. Зокрема, було відкрито порт 22 для SSH, що дозволяє безпечно підключатися до Raspberry Pi для керування системою через термінал. Окрім цього, для демонстрації роботи і підтримки веб-сервісів було дозволено вхідні HTTP-з'єднання на порт 80. Всі інші спроби встановити вхідні з'єднання були заборонені, що дозволило значно підвищити безпеку і уникнути потенційних атак через невідомі порти чи сервіси.

Ланцюг OUTPUT, відповідальний за вихідний трафік, налаштований на повний дозвіл передачі пакетів із системи, що забезпечує можливість фаєрволу ініціювати з'єднання і обмінюватися інформацією без обмежень.

Правила у ланцюзі FORWARD регулюють транзитний трафік, що проходить через фаєрвол між двома мережевими інтерфейсами. У нашому випадку eth1 підключений до внутрішньої локальної мережі, а eth0 – до зовнішнього Інтернету

через NAT VirtualBox. Було встановлено, що трафік з внутрішньої мережі може вільно проходити через фаєрвол до Інтернету, а у зворотному напрямку — тільки ті пакети, які належать до існуючих або пов'язаних з ними з'єднань (статуси RELATED, ESTABLISHED). Такий підхід забезпечує безпеку, оскільки дозволяє тільки відповідь на запити з локальної мережі, виключаючи несанкціоновані вхідні з'єднання.

Ще одним важливим елементом стала реалізація NAT (Network Address Translation), яка дозволяє багатьом пристроям у локальній мережі використовувати одну публічну IP-адресу для виходу в Інтернет. NAT реалізований через ланцюг POSTROUTING у таблиці nat, де встановлено правило MASQUERADE для інтерфейсу eth0. Це правило автоматично змінює адресу відправника у вихідних пакетах на IP-адресу інтерфейсу eth0, що дозволяє пристроям внутрішньої мережі "маскуватися" під адресу шлюзу при доступі до зовнішніх ресурсів.

Налаштування iptables здійснювалося з використанням команд у терміналі, що дозволило гнучко змінювати правила та оперативно реагувати на потреби тестування.

Завдяки такій конфігурації фаєрвол ефективно контролює і фільтрує трафік, забезпечуючи захист внутрішньої мережі, підтримку необхідних сервісів і організовуючи безпечний доступ до Інтернету для всіх пристроїв у локальній мережі.

Важливо також відзначити, що iptables є стандартним і широко підтримуваним інструментом у Linux, що гарантує сумісність, можливість масштабування і подальшого розширення функціоналу системи. Використання iptables в рамках Raspberry Pi OS дозволяє реалізувати ефективний і доступний за вартістю мережевий фаєрвол, який можна адаптувати під різні завдання і потреби.

Для наочності та підтвердження коректності налаштувань наводяться відповідні виводи команд, що демонструють активні правила та їх параметри.

Перевірка та тестування функціональності мережевого фаєрволу є критично важливим етапом у процесі його розробки та налаштування, оскільки саме він

дозволяє впевнитися у правильності реалізації правил безпеки та коректності обробки мережевого трафіку (рис.3.4 та 3.5). У нашому випадку тестування проводилося із залученням набору стандартних інструментів командного рядка Linux – ping, curl і netcat – які надають можливість як базової, так і розширеної діагностики мережевих з'єднань.

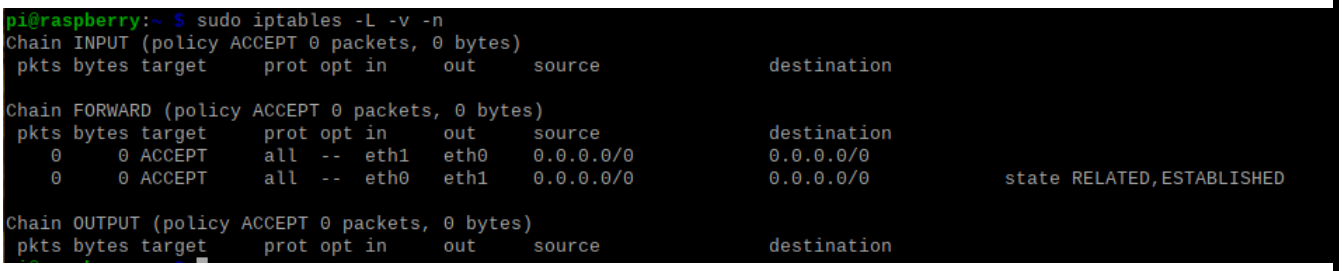


Рисунок 3.4 – Список усіх правил у ланцюгах FILTER з докладною статистикою

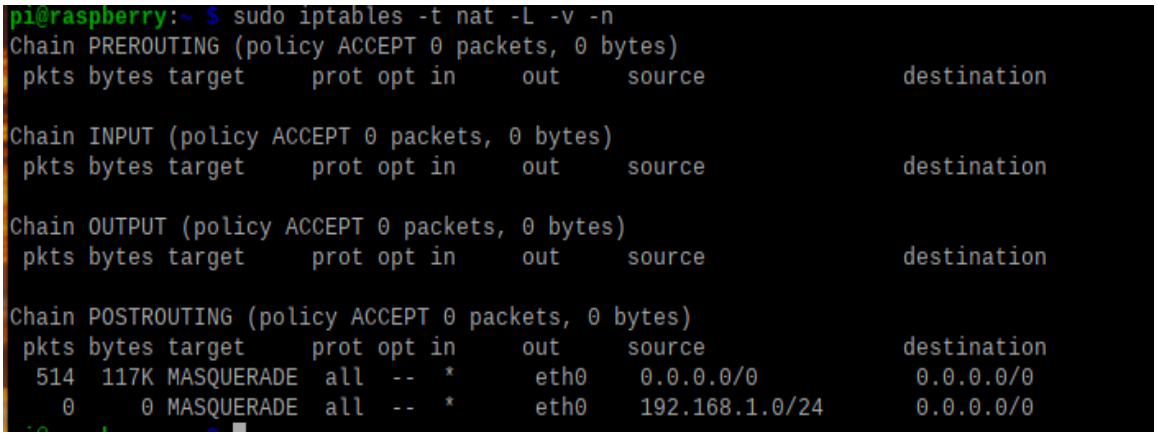


Рисунок 3.5 – Правила у таблиці NAT, зокрема правило маскараду для інтерфейсу eth0

Спершу було організовано тестове середовище, у якому клієнтська машина, що знаходиться у внутрішній мережі, була підключена до віртуального шлюзу – Raspberry Pi, що виконує функції фаєрволу. Така архітектура відтворює реальну мережеву інфраструктуру, де шлюз забезпечує контроль і фільтрацію трафіку, пропускаючи лише дозволені пакети і блокуючи підозрілі або небажані з'єднання.

Для перевірки базової зв'язності із зовнішніми ресурсами було використано утиліту ping, яка надсилає ICMP-запити до віддаленого хоста. У якості цільового

адресата було вибрано публічний DNS-сервер Google з IP-адресою 8.8.8.8. Виконання команди ping дозволяє переконатися, що мережеві пакети проходять через фаєрвол і коректно повертаються, що свідчить про працездатність маршрутизації та базову доступність Інтернет-з'єднання. У результаті тесту було зафіксовано стабільні відповіді від сервера з мінімальним часом затримки, що підтверджує якість мережевого зв'язку і правильність налаштувань NAT.

Наступним етапом було тестування можливості доступу до веб-ресурсів за допомогою утиліти curl. Це інструмент, який дозволяє надсилати HTTP-запити на вказані адреси і отримувати відповіді від вебсерверів. Використання curl допомогло перевірити, чи пропускаються запити на порт 80, який традиційно використовується для HTTP-трафіку. У тестах було здійснено кілька звернень до відомих веб-сайтів, і всі запити успішно отримували відповіді, що свідчить про коректну роботу правил фаєрволу, які дозволяють HTTP-трафік.

Крім того, перевірялося встановлення SSH-з'єднання з Raspberry Pi через порт 22, що є важливим для адміністративного доступу до пристрою. Тестування SSH продемонструвало, що дозволений доступ працює безперебійно, а всі інші спроби з'єднання, які не відповідали встановленим правилам, були заблоковані. Це підтвердило ефективність заходів безпеки щодо обмеження доступу до критичних сервісів.

Паралельно із перевіркою працездатності мережевих протоколів проводився аналіз логів системи за допомогою команд journalctl та dmesg, які дозволили відстежити записані повідомлення фаєрволу про проходження або блокування пакетів.

Ця інформація була цінною для підтвердження відповідності поведінки системи очікуваним правилам, а також для діагностики можливих помилок чи нестиковок у налаштуваннях. Логи фіксували всі спроби встановлення з'єднань, їх успішність або відмову, що дало змогу комплексно оцінити роботу системи.

Рисунок 3.6 ілюструє результат виконання команди ping до IP-адреси 8.8.8.8, де видно стабільні відповіді з часом затримки та відсутністю втрат пакетів. Рисунок

3.7 демонструє вивід curl із отриманням відповіді від вебсерверу, що підтверджує доступність HTTP-сервісів через фаєрвол.

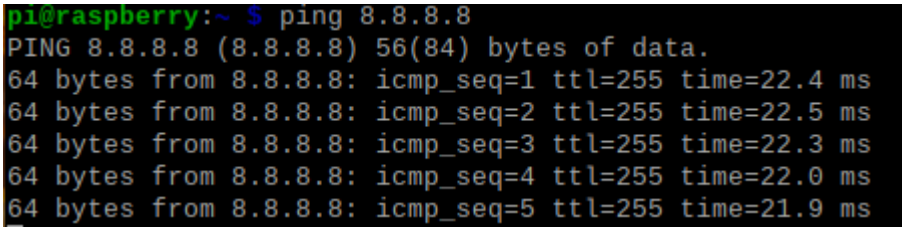


Рисунок 3.6 – Приклад пінгу 8.8.8.8

Приклад виконання пінгу до IP-адреси 8.8.8.8 показав, що мережеве з'єднання працює коректно. Відповіді на ICMP-запити надходили стабільно, без втрат пакетів, зі швидким часом відгуку, що свідчить про правильну маршрутизацію трафіку через фаєрвол і належне налаштування NAT. Це підтверджує, що внутрішня клієнтська машина має вільний доступ до зовнішніх ресурсів через захищений шлюз, а правила фільтрації не блокують необхідні мережеві протоколи. В цілому, результат пінгу демонструє ефективну роботу системи та її готовність до подальшого використання у реальних умовах.

Результат HTTP-запиту за допомогою інструмента curl підтвердив коректність налаштувань фаєрволу та мережевого з'єднання. Запит до веб-серверу був успішно виконаний, отримано відповідь з кодом 200 OK, що свідчить про стабільний доступ до зовнішніх HTTP-ресурсів. Це означає, що правила iptables правильно пропускають HTTP-трафік, не блокуючи легітимні запити користувачів. Відсутність помилок чи затримок в процесі обробки запиту також підтверджує належну роботу механізму NAT і маршрутизації. Отже, тест з curl демонструє, що фаєрвол ефективно забезпечує захист без перешкод для нормального функціонування мережі (рис. 37).

Проведене тестування підтвердило, що реалізований мережевий фаєрвол на базі Raspberry Pi OS із використанням iptables ефективно виконує поставлені завдання щодо контролю доступу, фільтрації трафіку та забезпечення безпеки локальної мережі. Водночас, результати тестування свідчать про високу

стабільність та надійність системи, що є важливим фактором для подальшого застосування розробленого рішення в реальних умовах.

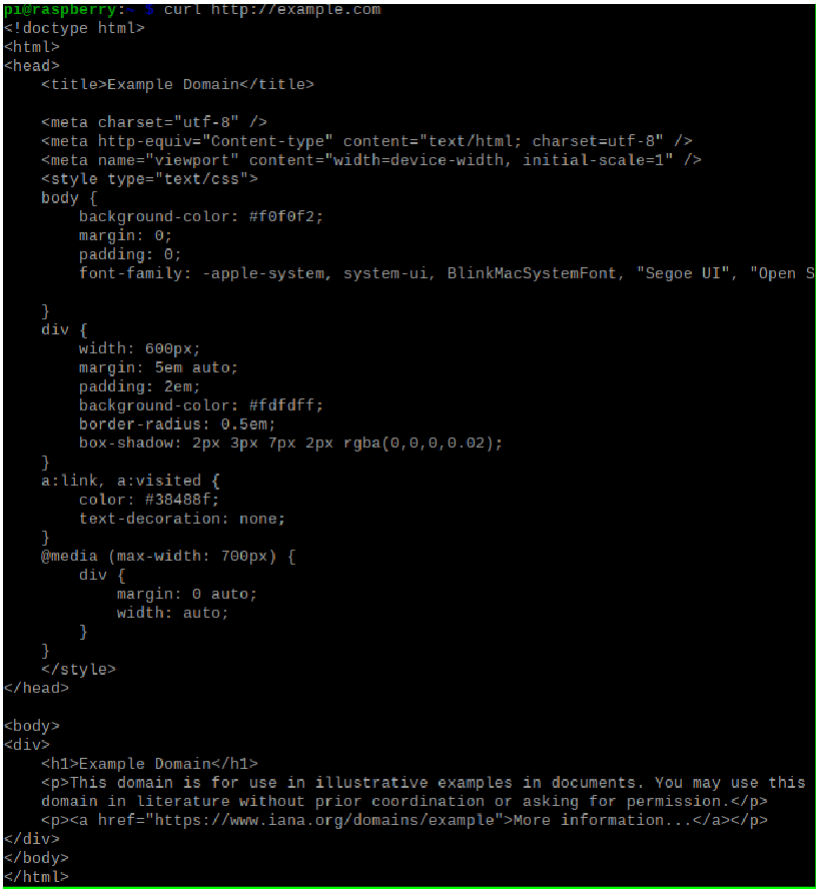


Рисунок 3.7 - Результат HTTP-запиту через curl

Резервне копіювання конфігурації iptables є важливим етапом у забезпеченні стабільності та надійності роботи мережевого фаєрволу. Після налаштування всіх необхідних правил доцільно зберегти їх у файл, щоб у разі збоїв, оновлення системи або перезавантаження мати можливість швидко їх відновити без потреби повторного налаштування вручну.

Щоб створити резервну копію, можна скористатися командою iptables-save, яка виводить поточні правила у вигляді текстового списку. Результат цієї команди зручно перенаправити у файл, наприклад у /etc/iptables/rules.v4 або будь-який інший зручний для користувача шлях. Такий файл можна зберігати як локально, так і на зовнішньому носії або у віддаленому сховищі.

Для відновлення конфігурації використовується команда iptables-restore, яка читає правила з раніше збереженого файлу та завантажує їх у пам’ять ядра, тим самим відновлюючи повноцінну роботу фаєрволу. Це особливо корисно при автоматичному застосуванні правил під час запуску системи – файл конфігурації може бути викликаний із відповідного скрипта або сервісу systemd.

Крім того, у випадку тривалого експериментування з правилами або внесення змін до структури мережі, резервна копія дає змогу повернутися до попередньо стабільного стану, що значно спрощує адміністрування та знижує ризик порушення безпеки через людський фактор.

Отже, реалізація регулярного резервного копіювання правил iptables та знання способів їх відновлення є необхідною складовою комплексного підходу до захисту мережевої інфраструктури, навіть у лабораторному чи емуляційному середовищі на базі VirtualBox з Raspberry Pi OS.

3.3 Перспективи вдосконалення та масштабування системи

Побудована у віртуальному середовищі система мережевого фаєрволу на базі Raspberry Pi OS з використанням інструменту iptables є лише базовим прототипом. Вона продемонструвала ключові принципи фільтрації трафіку, NAT, маршрутизації та захисту локальної мережі. Однак ця система має значний потенціал для вдосконалення і масштабування, як у напрямку функціонального розвитку, так і при впровадженні в реальне середовище.

Один із ключових напрямів удосконалення створеної системи полягає в її перенесенні з віртуального середовища VirtualBox на фізичну платформу – одноплатний комп’ютер Raspberry Pi. Такий підхід є логічним завершенням розробки, оскільки дозволяє вийти за межі лабораторного моделювання й протестувати систему в умовах реального функціонування. Підключення Raspberry Pi до фізичної мережі дає змогу перевірити її поведінку у справжньому середовищі

– з живим трафіком, мережевими подіями, перебоями живлення, змінами в конфігурації маршрутизаторів і клієнтських пристроїв.

На відміну від віртуального середовища, фізичне обладнання забезпечує реальний рівень взаємодії на апаратному рівні, що дозволяє точніше оцінити продуктивність системи, її стійкість до збоїв і здатність працювати у фоновому режимі протягом тривалого часу без перезавантаження. Raspberry Pi, завдяки своєму компактному форм-фактору, безвентиляторній конструкції та мінімальному енергоспоживанню, ідеально підходить для цілодобового використання. Його можна розмістити в будь-якому середовищі – як удома, так і в невеликому офісі – без потреби в охолодженні чи додатковому обслуговуванні.

Крім того, фізична реалізація дозволяє розширити функціонал: підключати додаткові модулі, зовнішні накопичувачі, використовувати USB-адаптери для створення додаткових мережевих інтерфейсів, або реалізувати резервне живлення для підвищення надійності. Використання реального пристрою також дає змогу інтегрувати фаєрвол у більші інфраструктурні рішення, наприклад, у систему моніторингу мережі, VPN-сервер або хмарний шлюз доступу.

Щоб зробити процес адміністрування мережевого фаєрволу зручнішим і доступнішим, доцільно реалізувати інтерфейс керування, який спростить взаємодію з системою для користувачів із різним рівнем технічної підготовки. Замість того, щоб редагувати правила iptables вручну через командний рядок, адміністратор матиме змогу користуватися інтуїтивним веб-інтерфейсом або спеціалізованою консольною панеллю.

У цьому контексті доцільно розглянути використання таких інструментів, як Webmin, Shorewall або Firewalld. Вони дозволяють візуально керувати правилами доступу, налаштовувати NAT, слідкувати за активними з’єднаннями, а також миттєво вносити зміни без потреби глибоко занурюватися в синтаксис iptables.

Впровадження графічного або структурованого текстового інтерфейсу керування особливо корисне для систем, які можуть обслуговуватися нефаховими користувачами, наприклад у малому офісі чи домашній мережі. Це відкриває

можливість делегувати частину рутинних задач некваліфікованому персоналу без ризику порушення основної конфігурації.

Окрім підвищення зручності керування, важливо також реалізувати елементи автоматизації. Наприклад, система має підтримувати регулярне збереження актуальної конфігурації фаєрволу у вигляді резервних копій, що дозволяє швидко відновити налаштування у разі збою, оновлення програмного забезпечення або випадкових змін. Автоматичне створення бекапів після кожного оновлення конфігурації гарантує збереження стабільної роботи системи навіть у непередбачуваних ситуаціях.

Крім цього, автоматизація може включати сценарії безперервного оновлення – оновлення системи без зупинки сервісів або порушення роботи мережі. Це критично важливо для фаєрволу, який працює у виробничому середовищі, де навіть короткочасна втрата доступу до Інтернету чи внутрішніх ресурсів може призвести до значних незручностей або втрат.

Загалом, розширення функціональності за рахунок інтерфейсу керування та механізмів автоматизації значно підвищує рівень зручності, безпеки та надійності розробленої системи. Вона стає не лише ефективним інструментом фільтрації трафіку, але й гнучким рішенням, яке легко підтримувати, розвивати та адаптувати під потреби конкретного користувача або організації.

Моніторинг і своєчасне реагування на інциденти безпеки є критично важливими аспектами управління будь-якою мережею, особливо коли мова йде про мережеві фаєрволи. Інтеграція спеціалізованих інструментів моніторингу дозволяє постійно відстежувати стан мережі, аналізувати її активність і виявляти потенційні загрози або нетипові ситуації на ранніх стадіях. Для цього можна використовувати як складні системи класу Zabbix, Prometheus у поєднанні з Grafana, так і простіші утиліти на кшталт vnstat, iftop чи logwatch. Вони дають змогу отримувати як детальну статистику про обсяг і напрямок трафіку, так і швидко помічати підвищену активність, що може свідчити про спроби несанкціонованого доступу, сканування портів чи інші атаки.

Особливо важливим є правильне налаштування логування фаєрволу через iptables. Записи про блоковані або дозволені з’єднання можуть накопичуватися локально, однак для більш глибокого аналізу та централізованого управління безпекою рекомендується переспрямовувати ці логи до спеціалізованих сховищ або SIEM-систем (Security Information and Event Management). Такі платформи дозволяють агрегувати, корелювати і аналізувати події з різних джерел, що сприяє виявленню складних атак, пошуку закономірностей у мережевій активності і формуванню ефективних заходів для запобігання загрозам у майбутньому.

Загалом, впровадження комплексного моніторингу з автоматичним оповіщенням про інциденти підвищує загальну стійкість мережевої інфраструктури. Це дозволяє оперативно реагувати на потенційні загрози і запобігати їхньому негативному впливу, забезпечуючи безперервність роботи та безпеку критично важливих сервісів.

Система може бути розширена рядом додаткових функцій і сервісів, які підвищать її можливості та адаптивність до конкретних вимог користувача або організації. Наприклад, доцільною є інтеграція з VPN-сервісами, такими як OpenVPN або WireGuard, що забезпечує захищений віддалений доступ до мережі, дозволяючи безпечно працювати з внутрішніми ресурсами з будь-якого місця. Це особливо актуально для організацій з розподіленою структурою або для тих, хто часто працює віддалено.

Для більш детальної фільтрації трафіку можна додати проксі-сервери, такі як Squid, або системи блокування реклами і шкідливих ресурсів, наприклад Pi-hole. Вони дають змогу контролювати доступ до конкретних доменів чи URL, підвищуючи рівень безпеки і продуктивність мережі за рахунок блокування небажаного контенту.

Також можливе впровадження систем виявлення вторгнень (IDS), таких як Snort або Suricata, які аналізують мережевий трафік на предмет підозрілої активності, дозволяючи оперативно виявляти і реагувати на потенційні загрози. Це дає змогу перейти від пасивного моніторингу до активного захисту мережі.

У випадку необхідності сегментації мережі для розділення різних підмереж або груп користувачів, можна налаштувати підтримку VLAN. Це дозволяє підвищити безпеку та оптимізувати трафік, виділяючи окремі логічні сегменти, що працюють незалежно один від одного, але об'єднані у єдину фізичну інфраструктуру.

Таким чином, розширення системи додатковими функціями і сервісами значно підвищує її гнучкість, безпеку і можливості управління, відповідаючи найвищим сучасним вимогам до мережевого захисту.

Коли мережа починає рости, а кількість підключених пристроїв збільшується, виникає потреба у масштабуванні системи мережевого захисту, щоб забезпечити стабільну роботу без зниження продуктивності. У цьому випадку можна організувати кластер фаєрволів – групу взаємодіючих пристроїв або віртуальних машин, які спільно обробляють мережевий трафік. Такий підхід дозволяє розподіляти навантаження між кількома вузлами, що запобігає перевантаженню окремого фаєрволу та підвищує загальну пропускну здатність системи.

Крім того, доцільно впровадити балансування навантаження, яке автоматично спрямовує трафік на менш завантажені вузли, підтримуючи рівномірний розподіл ресурсів. Це особливо корисно в офісах із значною кількістю користувачів або у випадках, коли обробка трафіку є критичною для бізнес-процесів.

Ще одним важливим аспектом є забезпечення високої доступності мережевого захисту. Для цього може бути реалізований резервний вузол, який знаходиться в режимі очікування і миттєво активується, якщо основний фаєрвол виходить з ладу через апаратні несправності, збій програмного забезпечення або інші причини. Ця резервна система забезпечує безперервність роботи мережі і мінімізує ризики простоїв, які можуть негативно вплинути на бізнес чи роботу користувачів.

Загалом, використання кластерів, балансування навантаження та резервних вузлів дозволяє створити стійку, масштабовану і надійну мережеву інфраструктуру, яка здатна ефективно працювати в умовах зростаючих вимог та забезпечувати безпеку навіть при підвищених навантаженнях.

Для забезпечення стабільної та безпечної роботи мережевого фаєрволу важливо регулярно оновлювати операційну систему, а також всі встановлені пакети, зокрема ті, що відповідають за безпеку. Оновлення допомагають усувати вразливості, які можуть бути використані зловмисниками для атак на систему. Окрім цього, підтримка актуальної версії ОС і програмних компонентів сприяє покращенню продуктивності і сумісності з новими технологіями.

Важливо також періодично переглядати та адаптувати правила iptables, щоб вони відповідали сучасним загрозам і вимогам безпеки. З часом можуть з’являтися нові типи атак, а мережеві умови можуть змінюватися, тому фаєрвол повинен бути гнучким і здатним швидко реагувати на ці зміни.

Для цього можна автоматизувати процеси оновлення чорних і білих списків IP-адрес, які визначають, який трафік слід блокувати або навпаки дозволяти. Завантаження актуальних списків шкідливих IP, відомих спамерів або джерел атак допомагає оперативно захищати мережу від небажаного трафіку. Крім того, застосування геофільтрації, що блокує підключення з певних регіонів, які не мають законного доступу до ресурсів, дозволяє знизити ризики проникнення зловмисників.

Регулярний аналіз логів, моніторинг активності та автоматичне коригування правил безпеки допомагають своєчасно виявляти потенційні загрози і усувати їх, підтримуючи стабільність і надійність роботи фаєрволу. Такий підхід забезпечує динамічний захист, адаптований під поточну ситуацію в мережі, і сприяє збереженню високого рівня безпеки.

Резервування та відновлення конфігурацій є критично важливим етапом у забезпеченні безперервної роботи мережевого фаєрволу. У разі програмних збоїв, помилок під час оновлень або випадкових змін у конфігурації система може

втратити працездатність, що спричинить перебої в роботі мережі, порушення захисту або повну втрату з’єднання з Інтернетом. Щоб мінімізувати ці ризики, необхідно впровадити механізми збереження актуальних конфігурацій і швидкого їх відновлення.

Зокрема, варто передбачити автоматизоване збереження конфігурацій iptables у вигляді файлів правил, які зберігаються в безпечному каталозі локально або на віддаленому сервері. Аналогічно, потрібно регулярно створювати резервні копії конфігурацій мережевих інтерфейсів, налаштувань NAT, DHCP-серверів (якщо використовуються), VPN-клієнтів або серверів, а також інших служб, які залучені до маршрутизації і захисту трафіку.

У реалізації цієї функціональності можуть бути використані скрипти на Bash або Python, які виконуватимуть щоденне або щотижневе збереження конфігурацій зі штампами часу. У разі збою адміністратор зможе обрати останню стабільну версію та швидко її відновити за допомогою простих команд або запуском автоматизованого скрипта. Додатково можна інтегрувати зберігання резервних копій до віддалених репозиторіїв — наприклад, через rsync, scp або git, що дозволить убезпечити дані навіть у разі втрати фізичного пристрою.

Варто також передбачити систему сповіщення про успішність або невдачу процесу резервування, щоб у разі збоїв адміністратор мав змогу оперативно відреагувати. Надсилання логів на пошту або через месенджери (наприклад, Telegram) може стати дієвим способом моніторингу процесу.

Реалізація надійної системи резервування конфігурацій не лише захищає від втрати критичних даних, а й значно полегшує тестування нових функцій, оскільки адміністратор завжди може швидко повернутися до стабільної версії конфігурації після експериментів або змін. Це сприяє гнучкості розвитку системи без шкоди для її стабільності.

Інтеграція з централізованими системами керування мережею (Network Management Systems, NMS) є важливим кроком у підвищенні ефективності адміністрування фаєрволів, особливо у випадках, коли мережа включає декілька

вузлів або коли рішення на базі Raspberry Pi планується масштабувати до рівня корпоративної або розподіленої інфраструктури.

Застосування систем централізованого управління, таких як Ansible, SaltStack або Puppet, дозволяє автоматизувати велику кількість рутинних операцій, пов'язаних із налаштуванням, оновленням та моніторингом стану пристроїв. Завдяки цим інструментам можливо організувати одночасне поширення конфігурацій iptables, зміни в політиках NAT, оновлення чорних списків або застосування патчів без необхідності вручну підключатися до кожного пристрою окремо.

Централізований підхід до управління забезпечує цілісність і уніфікацію мережевої політики безпеки, оскільки всі зміни здійснюються з одного контрольного вузла та можуть фіксуватися у вигляді шаблонів або playbook-файлів. Це дозволяє зберігати контроль над змінами, швидко відтворювати типові конфігурації на нових пристроях або відновлювати налаштування у випадку збою.

Крім того, сучасні NMS-платформи підтримують механізми зворотного зв'язку, що дає змогу отримувати інформацію про статус виконаних завдань, наявність помилок, завантаженість ресурсів, статус сервісів і загальний стан безпеки. Наприклад, у поєднанні з системами моніторингу (Prometheus, Zabbix тощо), можна не лише централізовано керувати пристроями, але й аналізувати події в реальному часі та реагувати на інциденти швидше й ефективніше.

Інтеграція з NMS дає особливі переваги в умовах зростання масштабів мережі або при використанні фаєрволів у віддалених локаціях, де фізичний доступ до Raspberry Pi обмежений або відсутній. Це значно знижує витрати на адміністрування та забезпечує надійність і оперативність управління розподіленими мережевими пристроями.

Мережева сегментація та створення гостьових зон є важливими заходами для підвищення рівня інформаційної безпеки та контролю за трафіком у локальній мережі. Поділ загальної інфраструктури на ізольовані підмережі дозволяє відмежувати критичні ресурси (сервери, внутрішні служби, робочі пристрої

персоналу) від зовнішніх або менш довірених елементів, таких як пристрої гостей, студентів або підрядників.

У випадку використання Raspberry Pi як шлюзу або фаєрволу можливо реалізувати кілька логічних зон мережі за допомогою VLAN (Virtual LAN) або окремих фізичних інтерфейсів. Наприклад, основна мережа може бути призначена для внутрішніх пристроїв, тоді як гостьова — для тимчасових підключень або відвідувачів. Розділення може базуватись також на рівні бездротової мережі шляхом створення окремого SSID для гостей із власною IP-підмережею.

Таке рішення дозволяє встановити окремі правила фільтрації трафіку через iptables для кожної з підмереж. Для гостьової зони можна обмежити доступ до внутрішніх IP-адрес, дозволивши лише вихід до Інтернету, блокування підозрілих портів, контроль швидкості передачі даних або використання проксі-сервісів для додаткового контролю.

Мережева сегментація також сприяє кращій масштабованості, полегшує діагностику проблем, підвищує ефективність моніторингу та спрощує впровадження політик безпеки на різних рівнях. У навчальних або корпоративних середовищах це дозволяє підтримувати порядок у мережевій структурі та зменшує ризик внутрішніх загроз або несанкціонованого доступу. Raspberry Pi, у поєднанні з правильно налаштованими маршрутизаторами, комутаторами та VLAN-підтримкою, може виступати в ролі недорогого, але ефективного шлюзу, здатного керувати такими ізольованими зонами.

3.4 Висновки до третього розділу

У третьому розділі було детально описано процес програмно-апаратної реалізації мережевого фаєрволу на базі одноплатної комп’ютерної системи Raspberry Pi із використанням віртуального середовища VirtualBox та операційної системи Raspberry Pi OS (i386). Завдяки застосуванню iptables вдалося побудувати ефективну систему фільтрації трафіку, яка забезпечує контроль вхідних, вихідних

та транзитних мережевих з’єднань, а також реалізувати функції NAT для організації доступу внутрішніх пристроїв до Інтернету. Моделювання локальної мережі з використанням двох мережевих інтерфейсів та адаптерів VirtualBox дозволило створити умови, максимально наближені до реальних, що сприяло якісному тестуванню й налагодженню правил фаєрволу. Проведені експерименти підтвердили коректну роботу системи, що забезпечує безпечне та контрольоване управління мережевим трафіком. Описані перспективи вдосконалення та масштабування системи відкривають можливості для подальшого розвитку, включаючи перенесення на фізичне обладнання Raspberry Pi, впровадження зручних інтерфейсів адміністрування, моніторингу, автоматизації, а також інтеграції додаткових сервісів безпеки. Загалом, реалізована система є гнучким і ефективним інструментом для забезпечення мережевого захисту в різних середовищах, що підтверджує її практичну цінність та перспективність.

Крім технічних аспектів, у третьому розділі також розкрито практичну значущість побудованого рішення в контексті сучасних вимог до інформаційної безпеки. У процесі реалізації було враховано не лише функціональність фаєрволу, а й зручність його розгортання та обслуговування, що є ключовими чинниками для малих офісів, навчальних закладів або домашніх користувачів з обмеженими ресурсами. Проєкт продемонстрував, що навіть за допомогою недорогих та енергоефективних рішень на базі Raspberry Pi можливо створити повноцінний елемент мережевої інфраструктури, здатний задовольнити базові потреби кіберзахисту. Така система може не лише виконувати роль шлюзу з фільтрацією трафіку, а й бути відправною точкою для подальшого розширення мережевої архітектури з додаванням VPN, IDS, проксі-сервера чи балансування навантаження. У підсумку, розділ демонструє, як на основі доступних засобів можна реалізувати потужне рішення для підвищення рівня безпеки та контролю в локальній мережі.

ВИСНОВКИ

У ході дослідження, проєктування та реалізації програмно-технічного засобу мережевого фаєрволу на базі одноплатного комп'ютера Raspberry Pi було доведено, що ця платформа є ефективною, доступною та гнучкою основою для побудови системи мережевого захисту локального рівня. Враховуючи постійне зростання кількості кіберзагроз, необхідність у бюджетних, проте надійних рішеннях для фільтрації трафіку та контролю доступу до мережевих ресурсів є надзвичайно актуальною – особливо для домашніх користувачів, навчальних закладів, невеликих офісів і організацій, що не мають змоги інвестувати в дорогі апаратні фаєрволи.

Raspberry Pi продемонстрував достатній рівень продуктивності для реалізації основних функцій фаєрволу, включаючи фільтрацію пакетів, трансляцію адрес (NAT), контроль вихідного та вхідного трафіку, а також підтримку таких важливих компонентів, як DNS/DHCP, захист від несанкціонованих доступів, ведення логів, аналіз трафіку тощо. Завдяки відкритому програмному забезпеченню, система базується на перевірених інструментах Linux-середовища – таких як iptables, nftables, dnsmasq, fail2ban, tcpdump – які забезпечують гнучкість налаштування та можливість розширення функціоналу відповідно до вимог користувача.

Було спроєктовано логічну архітектуру системи з двома фізичними або віртуальними мережевими інтерфейсами для розмежування зовнішнього (інтернет) та внутрішнього (локальна мережа) середовищ. Такий підхід забезпечив можливість реалізації повноцінного шлюзу, через який проходить увесь трафік, що в свою чергу дозволяє застосовувати політики безпеки на рівні мережевого ядра.

Під час практичної реалізації використовувалась емуляція середовища Raspberry Pi за допомогою VirtualBox, що дозволило протестувати функціональність, налаштування та стабільність системи до її фізичного впровадження. Це підтвердило ефективність використання віртуального середовища для розробки та попереднього тестування мережевих рішень.

Окрему увагу було приділено забезпеченню безпеки самої платформи Raspberry Pi. Проведено заходи із захисту SSH-доступу, реалізовано регулярне оновлення системи, налаштовано захист від підбору паролів та впроваджено політику резервного копіювання конфігураційних файлів, що забезпечує стабільну роботу системи впродовж тривалого часу.

Загалом, створена система фаєрволу на базі Raspberry Pi повністю задовольняє вимоги до мінімального рівня мережевої безпеки, забезпечуючи гнучке, масштабоване та доступне рішення, яке легко адаптується під конкретні умови експлуатації. Цей підхід може бути ефективно застосований не лише у навчальних цілях, а й у реальних умовах для захисту локальних мереж малого та середнього масштабу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Karthikeyan S. Raj R. A. Cruz M. V. Chen L. Vishal J. A. Rohith V. S. A systematic analysis on Raspberry Pi prototyping: Uses, challenges, benefits, and drawbacks. *IEEE Internet of Things Journal*. 2023. Vol.10(16). P. 14397–14417.

2. Karthika K. Dhanalakshmi S. Murthy S. M. Mishra N. Sasikala S. Murugan S. Raspberry Pi-enabled wearable sensors for personal health tracking and analysis. In: 2023 *International Conference on Self Sustainable Artificial Intelligence Systems* (ICSSAS). 2023. P. 1254–1259. IEEE.

3. Arreaga N. X. Enriquez G. M. Blanc S. Estrada R. Security vulnerability analysis for IoT devices Raspberry Pi using pentest. *Procedia Computer Science*. 2023. Vol. 224. P. 223–230.

4. Sajjan V. Sharma P. Analysis of air pollution by using Raspberry Pi-IoT. In: *2021 6th International Conference on Inventive Computation Technologies (ICICT)*. 2021. P. 178–183.

5. Sajjan V. Sharma P. Analysis of air pollution by using Raspberry Pi-IoT. In: *2021 6th International Conference on Inventive Computation Technologies (ICICT)*. 2021. P. 178–183.

6. Thothadri M. An analysis on clock speeds in Raspberry Pi Pico and Arduino UNO microcontrollers. *American Journal of Engineering and Technology Management*. 2021. Vol. 6(3). P. 41–46.

7. Khan B. M. Fahad M., Bilal R. Khan A. H. Performance analysis of Raspberry Pi 3 IP PBX based on Asterisk. *Electronics*. 2022. Vol. 11(20). Article No. 3313.

8. Kamath V. Renuka A. Performance analysis of the pretrained EfficientDet for real-time object detection on Raspberry Pi. In: *2021 International Conference on Circuits, Controls and Communications (CCUBE)*. 2021. P. 1–6.

9. Pratama N. P. Oktiawati U. Y. Analysis and implementation of Raspberry Pi based wireless access point and user access notification using Telegram. *Journal of Internet and Software Engineering*. Vol. 3(1). P. 1–11.

10. Rahmat H., Wahjuni S., Rahmawan H. Performance analysis of deep learning-based object detectors on Raspberry Pi for detecting melon leaf abnormality. *International Journal of Advanced Science, Engineering and Information Technology*. 2022. Vol. 12(2). P. 572–579.

11. Фото графік зростання кількості. URL: https://www.h-x.technology/ua/blog-ua/cyber-threats-forecast-2024-ua (дата звернення 30.04.2025)

12. Фото мережева схема – Інтернет → фаєрвол → внутрішня мережа. URL: https://tutorials.freshersnow.com/routers-and-firewall/ (дата звернення 30.04.2025)

13. Фото Raspberry Pi. URL: https://rozetka.com.ua/ua/388638099/p388638099/ (дата звернення 30.04.2025)

14. Фото порівняння характеристик Cisco ASA. URL: https://www.thenetworkdna.com/2023/01/ipsec-site-to-site-vpn-tunnel-between.html (дата звернення 30.04.2025)

15. Mathe S. E. Kondaveeti H. K. Vappangi S. Vanambathina S. D. Kumaravelu N. K. A comprehensive review on applications of Raspberry Pi. *Computer Science Review*. 2024. Vol. 52. Article No. 100636.

16. Gupta V. Panchal V. G. Singh V. Bansal D. Garg P. EmotionNet: ResNeXt inspired CNN architecture for emotion analysis on Raspberry Pi. In: *Proceedings of the 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*. 2021. P. 262–267. IEEE.

17. Jolles J. W. Broad-scale applications of the Raspberry Pi: A review and guide for biologists. *Methods in Ecology and Evolution*. 2021. Vol. 12(9). P. 1562–1579.

18. Bilash D. A. Analysis of methods of using the Raspberry Pi platform in the training of computer engineers.

19. Yantidewi M. Wati J. N. Deta U. A. Bibliometric analysis: Research trends of Raspberry Pi-based egg incubators in the last ten years (2013–2022). *Prisma Sains: Jurnal Pengkajian Ilmu dan Pembelajaran Matematika dan IPA IKIP Mataram*. 2023. Vol. 11(2). P. 307.

20. Worsley K. et al. Performance analysis of computer vision with machine learning algorithms on Raspberry Pi 3. in *Proc. Future Technologies Conf. (FTC)*. 2021. Vol. 1. P. 216–232.

21. Raman R. and Mehla A., Cognitive computing stress analysis in aging populations using cloud-connected SVM classifier solution with Raspberry Pi, in *Proc. 2024 Int. Conf. on Advances in Modern Age Technologies for Health and Engineering Science (AMATHE)*, 2024, pp. 1–5.

22. Ruiz E. Ortiz M. and Vinces L. A computational comparative analysis between Nvidia Jetson Nano and Raspberry Pi CM4 for the classification of white asparagus with SVM, in *Brazilian Technology Symposium*, 2021, pp. 506–513.

23. Mhamdi L. Dammak O. Cottin F. and Ben Dhaou I. Deep learning for COVID-19 contamination analysis and prediction using ECG images on Raspberry Pi 4, *Int. J. of Imaging Systems and Technology*, vol. 33, no. 6, pp. 1858–1869, 2023.

24. Bawane N. Giripunje S. Bawaskar A. and Nandgawali M. B. *Review on: Blood culture analysis using Raspberry Pi*. 2021. Vol. 1. P. 148-149.

25. Saathvika, R. Yahvi L. and Yamni R. Analysis of seed quality using deep learning in Raspberry Pi, *SGS-Engineering & Sciences*. vol. 1. no. 01. 2021.

26. Lee C. H. Lee W. H. and Kim S. M. Development of IoT-based real-time fire detection system using Raspberry Pi and fisheye camera, *Applied Sciences*, vol. 13, no. 15. Art. no. 8568. 2023.

27. Guvvala M. V. and Ch D., ThingSpeak based air pollution monitoring system using Raspberry Pi, in *Proc. 2023 3rd Asian Conf. on Innovation in Technology (ASIANCON)*. 2023. pp. 1–6.

28. Yang C. et al. A low-cost, ear-contactless electronic stethoscope powered by Raspberry Pi for auscultation of patients with COVID-19: Prototype development and feasibility study, *JMIR Medical Informatics*, vol. 9. no. 1. Art. no. e22753. 2021.

29. Mahmood S. et al. Evaluation of the Omni-secure firewall system in a private cloud environment, *Knowledge*, vol. 4. no. 2. pp. 141–170. 2024.

30. Ceragioli L. Degano P. and Galletta L. Can my firewall system enforce this policy?, *Computers & Security*. Vol. 117. Art. no. 102683. 2022.

31. Anwar R. W. Abdullah T. and Pastore F. Firewall best practices for securing smart healthcare environment: A review, *Applied Sciences*. Vol. 11. no. 19. Art. no. 9183. 2021.

32. Al-Haijaa Q. A. and Ishtaiwia A. Machine learning based model to identify firewall decisions to improve cyber-defense, *Int. J. of Advanced Science, Engineering and Information Technology*, vol. 11, no. 4, pp. 1688–1695, 2021.

33. Li B. Q. and Ma Y. L. A firewall effect during the rogue wave and breather interactions to the Manakov system, *Nonlinear Dynamics*, vol. 111, no. 2. pp. 1565–1575. 2023.

34. Dawadi B. R. Adhikari B. and Srivastava D. K. Deep learning technique-enabled web application firewall for the detection of web attacks, *Sensors*, vol. 23. no. 4. Art. no. 2073. 2023.

35. Sharma H. Next-generation firewall in the cloud: Advanced firewall solutions to the cloud, *ESP J. of Engineering & Technology Advancements (ESP-JETA)*, vol. 1. no. 1. pp. 98–111. 2021.

36. Tudosi A. D. Graur, A. Balan, D. G. and Potorac A. D. Design and implementation of a distributed firewall management system for improved security, in *Proc. 2023 22nd RoEduNet Conf. Networking in Education and Research (RoEduNet)*. 2023. pp. 1–6.

37. Tudosi A. D. Graur A. Balan D. G. and Potorac A. D. Research on security weakness using penetration testing in a distributed firewall, *Sensors*, vol. 23. no. 5. Art. no. 2683. 2023.

38. Sahu A. et al. Generation of firewall configurations for a large-scale synthetic power system, in *Proc. 2022 IEEE Texas Power and Energy Conf. (TPEC)*. 2022. pp. 1–6.

39. Togay C. Kasif A. Catal C. and Tekinerdogan B. A firewall policy anomaly detection framework for reliable network security, *IEEE Trans. on Reliability*, vol. 71, no. 1. pp. 339–347. 2021.

40. Farooq M. Khan, R. and Khan M. H. Stout implementation of firewall and network segmentation for securing IoT devices, *Indian J. of Science and Technology*, vol. 16. no. 33. pp. 2609–2621. 2023.

41. Alsaqour R. Motmi A. and Abdelhaq M. A systematic study of network firewall and its implementation, *Int. J. of Computer Science & Network Security*, vol. 21, no. 4, pp. 199–208. 2021.

42. Sinha M., Bera, P. and Satpathy M., An anomaly-free distributed firewall system for SDN, in *Proc. 2021 Int. Conf. on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*. 2021. pp. 1–8.

43. AL-Behadili H. N. K. Decision tree for multiclass classification of firewall access, *Int. J. of Intelligent Engineering and Systems*, vol. 14. no. 3. pp. 294–302 2021.

44. Wang S. Liu R. Guo, X. and Wei G. Design of web application firewall system through convolutional neural network and deep learning, in *Proc. 2022 Int. Conf. on Computers, Information Processing and Advanced Education (CIPAE)*. 2022. pp. 454–457.

45. Rahman M. H. et al. Machine learning approach on multiclass classification of internet firewall log files, in *Proc. 2023 Int. Conf. on Computational Intelligence and Sustainable Engineering Solutions (CISES)*. 2023. pp. 358–364.

46. Madhloom J. K. et al. An information security engineering framework for modeling packet filtering firewall using neutrosophic Petri nets, *Computers*, vol. 12, no. 10. Art. no. 202. 2023.

47. Kovačević I. Štengl B. and Groš S. Systematic review of automatic translation of high-level security policy into firewall rules, in *Proc. 2022 45th Jubilee Int. Conv. on Information, Communication and Electronic Technology (MIPRO)*. 2022. pp. 1063–1068.

48. Prasetyo S. E. Haeruddin H., and Ariesryo K. Website security system from denial of service attacks, SQL injection, cross-site scripting using web application firewall, *Antivirus: Jurnal Ilmiah Teknik Informatika*, vol. 18. no. 1. pp. 27–36. 2024.

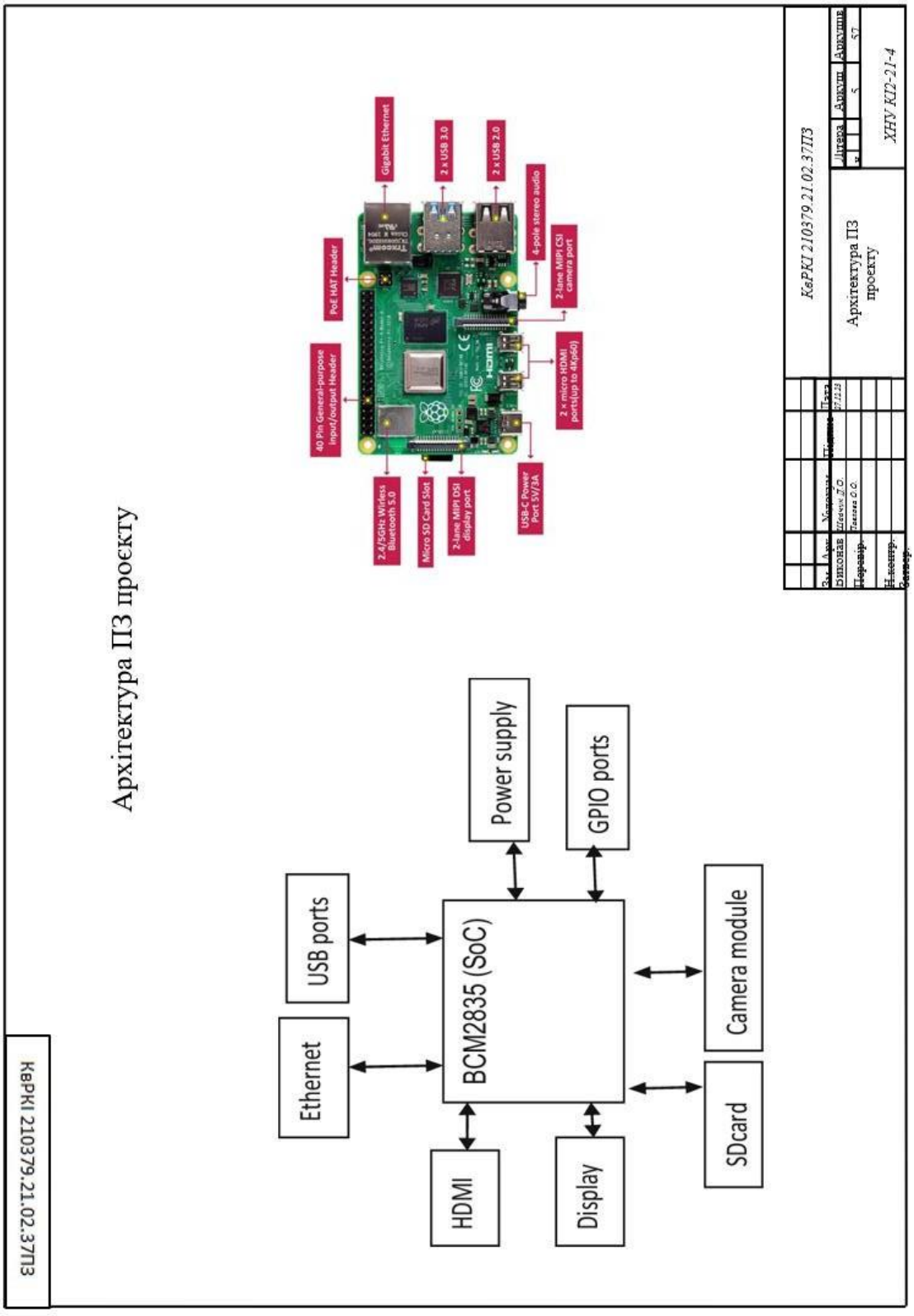
49. Yao H. et al. ControlNET: A firewall for RAG-based LLM system, *arXiv preprint*. arXiv:2504.09593. 2025.

50. Hakani D. A survey on firewall for cloud security with anomaly detection in firewall policy, in *Proc. 2023 Int. Conf. on Artificial Intelligence and Smart Communication (AISC)*. 2023. pp. 825–830.

Додаток А

(обов'язковий)

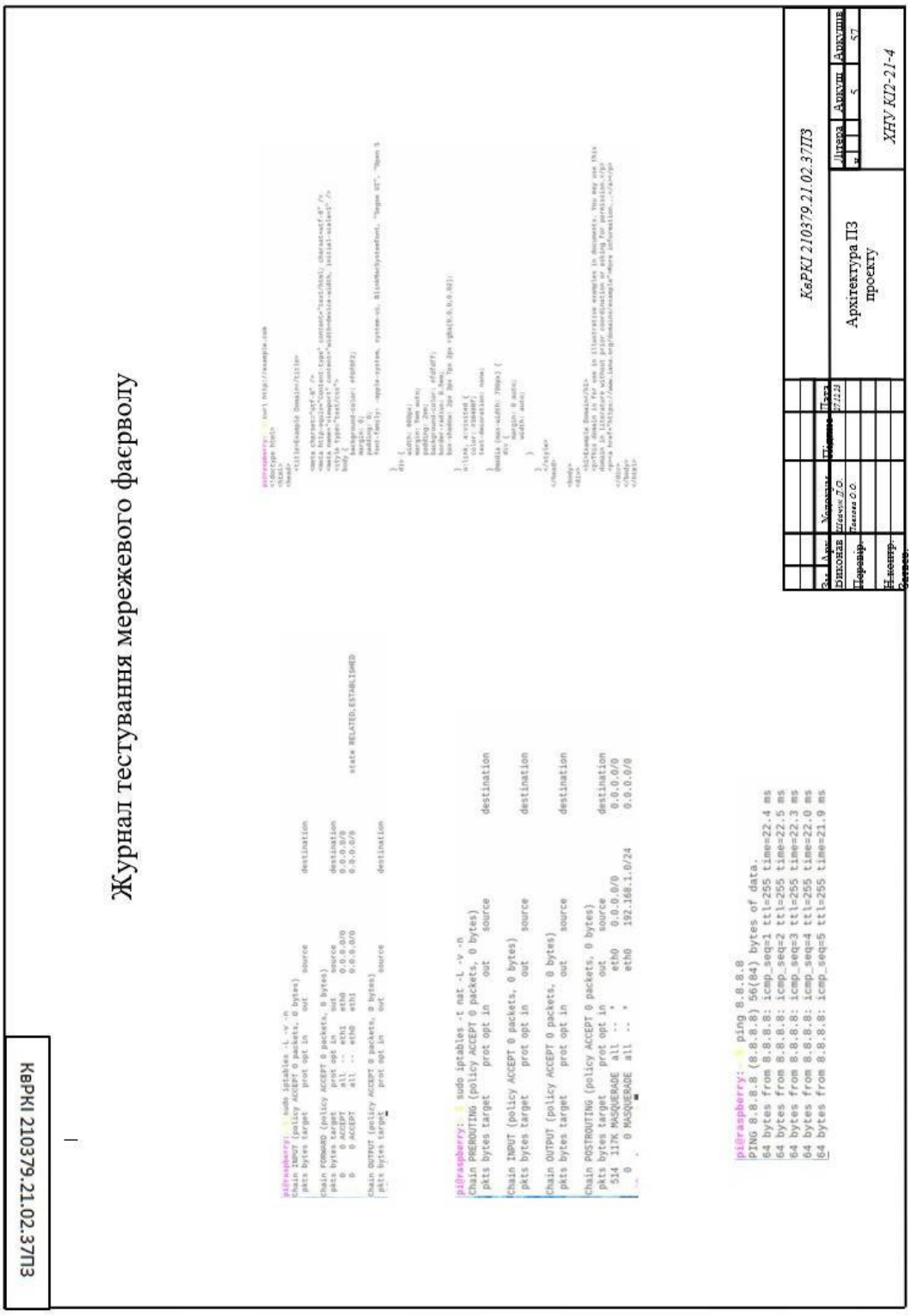
КОПІЯ КРЕСЛЕННЯ «АРХІТЕКТУРА ПЗ ПРОЄКТУ»



Додаток Б

(обов'язковий)

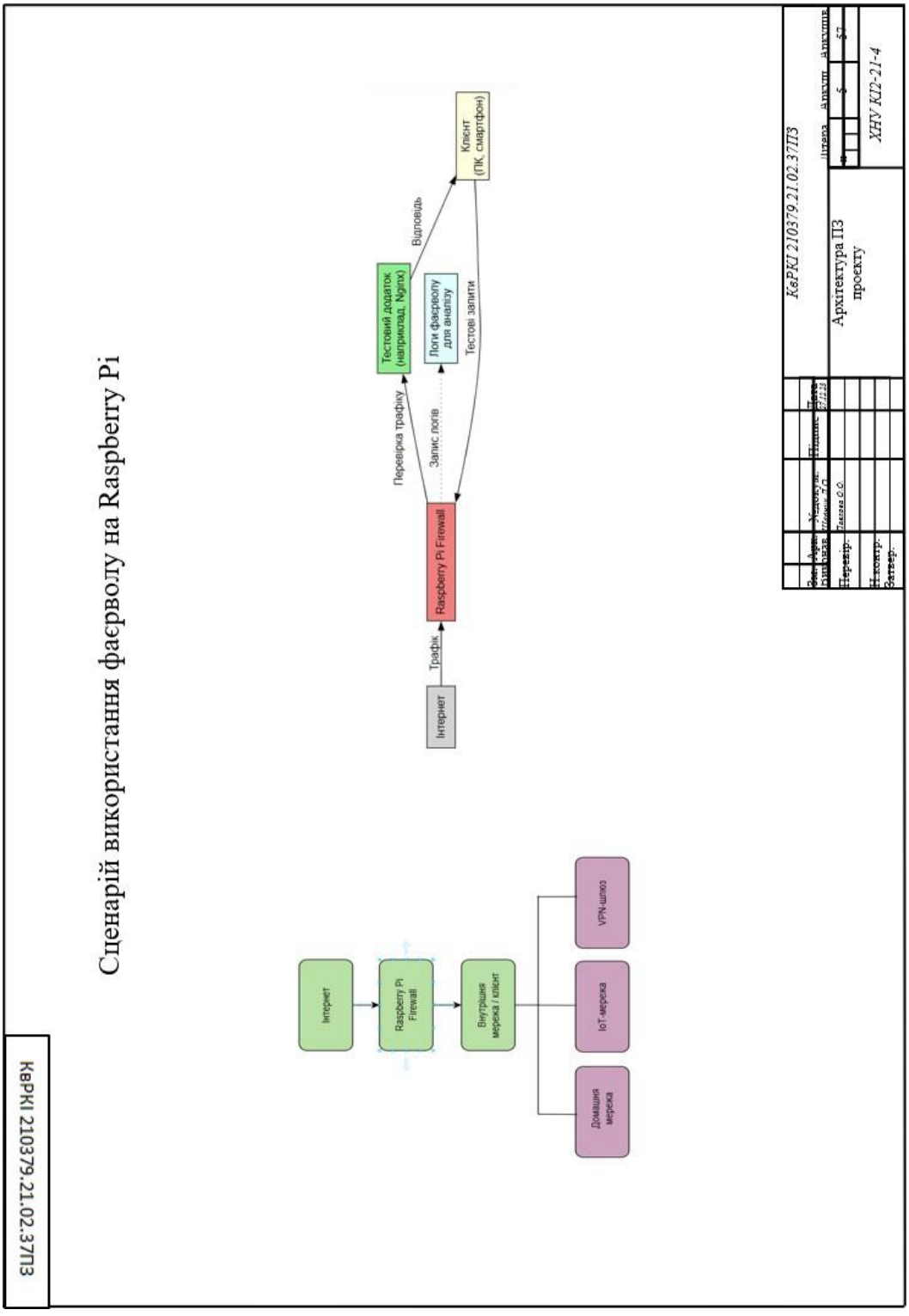
КОПІЯ КРЕСЛЕННЯ «ЖУРНАЛ ТЕСТУВАННЯ МЕРЕЖЕВОГО ФАЄРВОЛУ»



Додаток В

(обов’язковий)

СЦЕНАРІЙ ВИКОРИСТАННЯ ФАЄРВОЛУ НА RASPBERRY PI



Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 11.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 15%**

ID: 242961 Title: БКР Програмно технічні засоби мережевого фаєрволу на основі одноплатної компютерної системи Raspberry Pi Added in a DB: 2025-06-02 Authors: Денис ШЕВЧУК Heads: Сергій ЛИСЕНКО Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	100253	693	12008 (12%)	95 (14%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes
240777	Title: Звіт з ПДП Кіберфізична система моніторингу автоперевізників на основі Arduino Added in a DB: 2025-05-02 Authors: Д.О. Шевчука Heads: Лисенко С.М. Consultants: Opponents:	11161 (11.0%)	86 (12.0%)

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Денис ШЕВЧУК

Співавтор:

Назва: Шевчук_Програмно технічні засоби мережевого фаєрволу на основі одноплатної компютерної системи Raspberry Pi

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:5.1%

Коефіцієнт подібності 2:1.4%

Мікропробіли: 6

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-02 23:02:19.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2025-06-03 Доцент Андрій Нічепорук

Дата експерт

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Шевчук Денис Олександрович

Тема: Програмно-технічні засоби мережевого фаєрволу на основі одноплатної компютерної системи Raspberry Pi

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень ___3___ Кількість сторінок записки ___55___

1. Метою кваліфікаційної роботи є дослідження підходів до розгортання та конфігурації мережевого фаєрволу на базі одноплатного комп'ютера Raspberry Pi.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. У першому розділі було проаналізовано предметну область побудови мережевих фаєрволів з акцентом на використання одноплатних комп'ютерів Raspberry Pi як платформи для створення бюджетних рішень у сфері мережевої безпеки. Розглянуто принципи роботи фаєрволів, типи засобів захисту, а також проведено порівняльний аналіз сучасних програмних та апаратних рішень, зокрема iptables, nftables, pfSense, MikroTik, Cisco ASA. Визначено основні вимоги до мережевих фаєрволів: ефективна фільтрація трафіку, простота налаштування, масштабованість, низьке енергоспоживання та вартість.

У другому розділі здійснено технічне проєктування фаєрволу на основі Raspberry Pi. Запропоновано архітектуру системи, у якій Raspberry Pi виконує роль шлюзу між зовнішньою мережею (Інтернет) і внутрішніми клієнтами. Було детально описано способи розгортання iptables та nftables у середовищі Raspberry Pi OS (Raspios Bullseye), розглянуто типові сценарії використання — домашня мережа, офіс, IoT-інфраструктура, VPN-шлюз тощо. Враховано передові підходи до конфігурації політик доступу, NAT, фільтрації пакетів, а також інтеграцію з іншими сервісами захисту.

У третьому розділі виконано практичну реалізацію фаєрволу на базі Raspberry Pi

в емуляційному середовищі VirtualBox. Для тестування було використано Raspberry Pi Desktop (i386) як платформу, на якій налаштовано базову систему захисту з iptables та nftables. Розроблено й апробовано правила фільтрації трафіку, які демонструють типові сценарії захисту: блокування вхідних підключень, дозволені порти, NAT-маскування, журналювання подій. Реалізовано просту схему маршрутизації, у якій Raspberry Pi виступає в ролі шлюзу з подвійним мережевим інтерфейсом. Продемонстровано, що навіть з обмеженими апаратними ресурсами Raspberry Pi може ефективно виконувати функції фаєрволу у малих мережах. Усі налаштування базуються на відкритому програмному забезпеченні з використанням сучасних підходів до безпеки.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: 32-бітна система яка не підтримує всі рішення.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре (4.25)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Радевчук Галина Іванівна, доцент кафедри АКІТ та Р ХНУ канд. техн. наук

"05" 06 2025 р. _____ (підпис)

Завідувачу кафедри КІІС
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Дениса ШЕВЧУКА
ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-4

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

05.06 2025 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-технічні засоби мережевого фаєрволу на основі одноплатної компютерної системи Raspberry Pi

Автор: Денис ШЕВЧУК

Спеціальність: 123– Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Сергій ЛИСЕНКО, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
4) в якості запозичень в окремих місцях системою зафіксовано послідовності чотирьохрозряних двійкових кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 5.13% і адресується до 47 першоджерела; та системою Anti-Plagiarism складає 11%, що, з урахуванням наведених обгрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Сергій ЛИСЕНКО

Гарант ОП _____ Андрій Нічепорук

Завідувач кафедри КІІС _____ Ольга ПАВЛОВА