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

КВАЛІФІКАЦІЙНА РОБОТА

Апаратно-програмний засіб для захисту IoT-пристроїв від DDoS-атак

Назва теми

Рівень вищої освіти *перший (бакалаврський)*

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

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

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

Шифр КвРКІ.2301104.23.01.14 ПЗ

Виконав здобувач ІІІ курсу, група КІ2с-23-1 _____ Назар ДЗЮБАК
Підпис Ініціали, прізвище

Керівник _____ Олег САВЕНКО
Науковий ступінь, учене звання Підпис Ініціали, прізвище

Нормоконтролер _____ Сергій ЛИСЕНКО
Науковий ступінь, учене звання Підпис Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІІС _____ Ольга ПАВЛОВА
«04» _червня_ 2026 р. підпис Ініціали, прізвище
дата

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ
Завідувачка кафедри КІІС
Ольга ПАВЛОВА

" 10 " 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дзюбаку Назару Вікторовичу
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Апаратно-програмний засіб для захисту IoT-пристроїв від DDoS-атак

Керівник проекту (роботи) Савенко Олег Станіславович, д.т.н., проф.
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

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

Апаратно-програмний засіб для захисту IoT-пристроїв від DDoS-атак та постановка задачі щодо її удосконалення

Проєктування апаратно-програмного засобу для захисту IoT-пристроїв від DDoS-атак

Апаратно-програмна реалізація системи апаратно-програмного засобу для захисту IoT-пристроїв від DDoS-атак

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

Архітектура ПЗ проєкту

Діаграма взаємодії з користувачем

Апаратне забезпечення проєкту

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийн

7. Дата видачі завдання « 10 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Приміт
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	виконан
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконан
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2026	виконан
4	Робота над розділом 2 – вибір компонентів для проєктування апаратно-програмного засобу для захисту IoT-пристроїв від DDoS-атак	01.04.2026	викона
5	Робота над розділом 3 – проєктування апаратно-програмного засобу для захисту IoT-пристроїв від DDoS-атак	29.04.2026	виконан
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	викон
7	Попередній захист ВКР	26.05.2026	викон
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач _____ Назар ДЗЮБАК
Підпис Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи _____ Олег САВЕНКО
Підпис Ім'я, ПРІЗВИЩЕ

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1		КвРКІ 2301104.23.01.14 ПЗ	Пояснювальна записка	67		
			Графічні матеріали			
2		КвРКІ 2301104.23.01.14 Е8	Архітектура ПЗ проєкту	1		
3		КвРКІ 2301104.23.01.14 Е8	Діаграма взаємодії з користувачем	1		
4		КвРКІ 2301104.23.01.14 Е8	Апаратне забезпечення проєкту	1		

КвРКІ 2301104.23.01.14 ВП

Зм	Арк	№ докум	Підпис	Дата
Розробив		Дзюбак		
Перевір.		Савенко		
Н. контр.		Лисенко		
Затв.		Павлова		

Апаратно-програмний засіб для захисту IoT-пристроїв від DDoS-атак

Відомість проекту

Літера	Аркуш	Аркушів
У	1	1

ХНУ, КІ2с-23-1

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Апаратно-програмний засіб для захисту IoT-пристроїв від DDoS-атак».

Автор роботи: Назар ДЗЮБАК.

Керівник роботи: Олег САВЕНКО.

Пояснювальна записка: 67 с., 15 рис., 3 дод., 49 джерел.

Графічна частина: 3 креслення.

АЛГОРИТМ, АПАРАТНО-ПРОГРАМНИЙ ЗАСІБ, DDOS-АТАКА, IOT-ПРИСТРІЙ, МЕРЕЖЕВИЙ ТРАФІК, RASPBERRY PI, ФІЛЬТРАЦІЯ.

Кваліфікаційна робота бакалавра присвячена розробці та перевірці апаратно-програмного засобу для захисту IoT-пристроїв від DDoS-атак на базі одноплатного комп'ютера Raspberry Pi. Актуальність теми зумовлена зростанням кількості підключених пристроїв Інтернету речей, які використовуються в побутових, навчальних, офісних та виробничих середовищах, але часто мають обмежені апаратні ресурси, спрощені механізми безпеки та недостатній захист від аномального мережевого навантаження.

Метою роботи є проєктування, реалізація та перевірка апаратно-програмного засобу для виявлення аномального мережевого трафіку й фільтрації підозрілих пакетів у локальному IoT-сегменті. Для досягнення поставленої мети виконано аналіз особливостей побудови IoT-мереж, розглянуто основні види DDoS-активності, визначено ознаки аномального трафіку, проаналізовано існуючі засоби захисту, обґрунтовано вибір Raspberry Pi як центрального фільтрувального вузла, розроблено загальну архітектуру засобу, структуру мережевого фільтра, алгоритм виявлення аномальної активності та механізм реагування на підозрілий трафік.

Підпис здобувача

30.05.2026

Дата

ЗМІСТ

ВСТУП

Розвиток Інтернету речей суттєво змінив підхід до побудови сучасних комп'ютерних, вбудованих і мережевих систем. Якщо раніше до мережевої інфраструктури переважно підключалися персональні комп'ютери, сервери, мережеві принтери та інше відносно потужне обладнання, то сьогодні значну частину підключених вузлів становлять малоресурсні IoT-пристрої.

Особливість IoT-пристроїв полягає в тому, що вони часто мають обмежені апаратні ресурси, спрощену програмну логіку та мінімальний набір вбудованих механізмів захисту. Багато таких пристроїв орієнтовані передусім на виконання однієї прикладної функції: зчитування показників, передавання телеметрії, керування виконавчим елементом або відображення стану певного об'єкта.

Однією з найбільш небезпечних загроз для IoT-інфраструктури є DDoS-атаки. Їхня сутність полягає у створенні надмірного потоку мережевих пакетів або запитів, які надходять до цільового вузла з одного або багатьох джерел. Метою такої атаки є порушення доступності пристрою, сервісу або мережевого сегмента. Для потужних серверних систем DDoS-атака становить серйозну проблему, однак для IoT-пристроїв вона є ще небезпечнішою, оскільки навіть відносно невелике навантаження може призвести до затримок, зависання, втрати зв'язку або перезавантаження. Унаслідок цього пристрій перестає виконувати свою основну функцію, а вся система може втратити частину працездатності.

Актуальність теми бакалаврської роботи зумовлена потребою у створенні доступного локального засобу захисту IoT-пристроїв від аномального мережевого трафіку. Потужні хмарні сервіси та провайдерські системи захисту здатні протидіяти великим DDoS-атакам, але вони не завжди є зручними або виправданими для невеликої локальної IoT-мережі. Для навчального, домашнього або малого офісного середовища більш доцільним є компактне апаратно-програмне рішення, яке можна розмістити безпосередньо перед IoT-сегментом. Саме таким рішенням у роботі виступає фільтр на базі Raspberry Pi,

який аналізує мережевий трафік, визначає ознаки аномальної активності та застосовує правила обмеження або блокування підозрілих пакетів.

Метою бакалаврської роботи є розроблення апаратно-програмного засобу для захисту IoT-пристроїв від DDoS-атак, який на базі Raspberry Pi забезпечує виявлення аномального мережевого трафіку, фільтрацію підозрілих пакетів, журналювання подій і відображення стану системи. Досягнення цієї мети передбачає не лише теоретичний аналіз проблеми, а й формування практичного прототипу, здатного працювати в умовах локального експериментального стенда.

Об’єктом бакалаврської роботи є процес захисту IoT-пристроїв від аномального мережевого навантаження в локальному мережевому середовищі. У цьому процесі головну увагу приділено збереженню доступності пристроїв, своєчасному виявленню підозрілої активності та зменшенню впливу надмірного трафіку на малоресурсні вузли. Предметом бакалаврської роботи є апаратно-програмний засіб на базі Raspberry Pi, який виконує роль локального фільтра, аналізує параметри мережевого трафіку, визначає ознаки DDoS-активності та застосовує правила реагування.

Для досягнення поставленої мети в роботі передбачено виконання кількох взаємопов’язаних завдань. Насамперед проаналізовано особливості побудови та функціонування IoT-мереж, оскільки саме вони визначають вимоги до засобу захисту. Далі розглянуто загальну характеристику DDoS-атак на IoT-інфраструктуру та визначено, чому такі атаки є небезпечними для пристроїв з обмеженими ресурсами. Окрему увагу приділено ознакам аномального мережевого трафіку, зокрема різкому збільшенню кількості пакетів, надмірній активності окремих IP-адрес, великій кількості TCP SYN-пакетів, UDP-пакетів, ICMP-запитів і звернень до нетипових портів.

1 АНАЛІЗ ПРОБЛЕМИ ЗАХИСТУ ІОТ-ПРИСТРОЇВ ВІД DDOS-АТАК

1.1 Особливості побудови та функціонування IoT-мереж

Інтернет речей є одним із найпомітніших напрямів розвитку сучасних комп'ютерних і вбудованих систем, оскільки він поєднує фізичні пристрої, мережеві технології, програмне забезпечення та засоби оброблення даних в єдине середовище. У такій мережі звичайні технічні об'єкти отримують можливість передавати інформацію, приймати команди, реагувати на зміну зовнішніх умов і взаємодіяти з іншими пристроями без постійного ручного керування. До IoT-пристроїв належать датчики температури, вологості, освітленості, руху, газу, камери відеоспостереження, розумні розетки, контролери освітлення, мікроконтролерні вузли, побутові пристрої, елементи систем доступу та інше обладнання, яке має мережеве підключення. У межах бакалаврської роботи IoT-мережу розглянуто як сукупність апаратних вузлів, каналів зв'язку, програмних сервісів і засобів керування, які забезпечують збирання, передавання, оброблення та використання даних у локальному або віддаленому середовищі [6, 28].

На відміну від класичних комп'ютерних мереж, де основними елементами часто виступають персональні комп'ютери, сервери, комутатори та маршрутизатори, IoT-мережі мають значно більш різнорідну структуру. В одному сегменті можуть одночасно працювати прості сенсорні модулі, мікроконтролери, IP-камери, шлюзи, виконавчі пристрої, мобільні застосунки, локальні сервери та хмарні платформи. Кожен такий компонент має власне призначення, власні апаратні можливості та власний спосіб взаємодії з мережею. Через це IoT-середовище складніше контролювати, ніж звичайну локальну мережу з однотипними робочими станціями. Частина пристроїв може працювати на повноцінній операційній системі, а частина - лише на спрощеній мікропрограмі, яка виконує обмежений набір дій [14, 39].

Однією з головних особливостей IoT-пристроїв є обмеженість ресурсів. Багато вузлів мають невеликий обсяг оперативної пам'яті, слабкий процесор, обмежену енергоефективну архітектуру та мінімальний набір вбудованих засобів захисту. Для них основним завданням є не складна обробка мережевого трафіку, а зчитування показників, виконання керуючої дії або передавання короткого повідомлення. Через це такі пристрої не завжди здатні самостійно аналізувати вхідні підключення, виявляти підозрілі пакети, формувати правила блокування або відстежувати надмірну кількість запитів. Саме ця обмеженість робить IoT-сегмент вразливим до перевантаження, особливо у випадках, коли на пристрій спрямовується велика кількість однотипних мережевих звернень [3, 21].

Функціонування IoT-мережі також залежить від адресації та маршрутизації. У локальному середовищі пристрої часто отримують IP-адреси через DHCP, після чого взаємодіють із маршрутизатором, локальним сервером або хмарним сервісом. Частина вузлів може мати статичні адреси, особливо якщо вони виконують важливу функцію моніторингу, керування або відеоспостереження. Для фільтрації трафіку це має важливе значення, оскільки система повинна розуміти, які адреси належать захищеним IoT-пристроям, які адреси належать адміністративним вузлам, а які є зовнішніми джерелами. Без такого поділу складно коректно визначити, чи є потік даних нормальним, службовим або потенційно шкідливим [15, 43].

Побудова IoT-мережі з використанням окремого фільтра дозволяє сформувати більш кероване середовище. У такому середовищі кожен пакет, що надходить до захищеного сегмента, може бути врахований, класифікований і порівняний із заданими ознаками нормальної поведінки. Якщо активність відповідає звичайному режиму, трафік пропускається до пристрою. Якщо показники різко перевищують допустимі межі, фільтр може позначити джерело як підозріле, записати подію до журналу, сформувати правило блокування або

обмежити інтенсивність запитів. Це дає змогу зменшити навантаження на IoT-пристрій ще до того, як шкідливий трафік потрапить до нього напряму [16, 45].

У межах бакалаврської роботи особливості побудови та функціонування IoT-мереж мають безпосередній зв'язок із подальшим розробленням апаратно-програмного засобу. Обмежені ресурси пристроїв пояснюють потребу у винесенні захисної логіки на Raspberry Pi. Передбачуваний характер звичайного IoT-трафіку дає змогу застосовувати пороговий аналіз і виявляти різкі відхилення. Багаторівнева структура мережі показує, що захист має працювати не лише на рівні окремого пристрою, а й на рівні взаємодії між сегментами. Через це IoT-мережа в цій роботі розглядається не просто як набір підключених пристроїв, а як середовище, у якому потрібно забезпечити контроль трафіку, своєчасне виявлення аномалій і збереження доступності захищених вузлів [4, 31].

1.2 Загальна характеристика DDoS-атак на IoT-інфраструктуру

DDoS-атака є одним із найбільш поширених видів мережевих атак, спрямованих на порушення доступності інформаційної системи, сервісу або окремого пристрою. Її сутність полягає у створенні надмірного потоку запитів або пакетів, які надходять до цільового вузла з багатьох джерел одночасно. На відміну від звичайної DoS-атаки, де навантаження формується переважно з одного джерела, DDoS-атака має розподілений характер. У ній можуть брати участь десятки, сотні або тисячі заражених пристроїв, які разом створюють потік трафіку, що перевищує можливості оброблення цільової системи. Для IoT-інфраструктури така загроза є особливо небезпечною, оскільки більшість пристроїв не має достатнього запасу обчислювальних ресурсів і не розрахована на оброблення великої кількості одночасних мережевих звернень [7, 29].

Основною метою DDoS-атаки є не обов'язково викрадення даних або отримання несанкціонованого доступу, а саме виведення системи з нормального

режиму роботи. У випадку IoT-мережі це може проявлятися у втраті зв'язку з датчиками, зависанні контролерів, недоступності вебінтерфейсу, затримках передавання телеметрії, збої роботи виконавчих механізмів або повному припиненні обміну даними між пристроями. Якщо йдеться про побутову систему, наслідком може стати тимчасова недоступність камери, розумної розетки або датчика. Якщо ж IoT-інфраструктура використовується у виробничому, охоронному, транспортному чи енергетичному середовищі, навіть коротке порушення доступності може мати значно серйозніші наслідки [12, 41].

У межах бакалаврської роботи особливо важливими є ті види DDoS-активності, які можна змоделювати в лабораторному середовищі та виявити за допомогою локального фільтра. До таких проявів належить підвищена кількість TCP SYN-пакетів, надмірний UDP-трафік, часті ICMP-запити, велика кількість звернень до одного IoT-вузла та повторювані підключення з одного джерела. Для їхнього виявлення не обов'язково застосовувати складні методи машинного навчання. У межах практичного прототипу достатньо реалізувати пороговий або комбінований підхід, за якого система порівнює поточні показники трафіку з допустимими межами та формує подію безпеки у разі перевищення [4, 36].

Виявлення DDoS-активності в IoT-мережі ускладнюється тим, що не кожне збільшення трафіку є атакою. Наприклад, оновлення прошивки, передавання відеопотоку, підключення нового пристрою або активне використання вебінтерфейсу також може збільшити кількість пакетів. Через це засіб захисту має не лише рахувати загальний обсяг трафіку, а й враховувати його тип, напрям, джерело, тривалість і повторюваність. Якщо зростання трафіку є короткочасним і відповідає очікуваній дії, воно не обов'язково повинне сприйматися як атака. Якщо ж висока активність триває протягом декількох часових інтервалів, надходить з підозрілих адрес або має однотипну структуру, вона може розглядатися як аномальна [13, 31].

У практичній реалізації засобу захисту важливу роль відіграє часовий аналіз. Трафік можна оцінювати не як суцільний потік, а як набір показників за

короткі проміжки часу. Наприклад, система може визначати кількість пакетів за одну секунду, кількість SYN-пакетів за п’ять секунд, кількість UDP-пакетів за певний інтервал або кількість звернень від одного джерела до конкретного порту. Такий підхід дозволяє швидше реагувати на різкі зміни та не чекати повного перевантаження пристрою. Для IoT-середовища це особливо важливо, оскільки запас продуктивності в пристроїв невеликий, а затримка в реакції може призвести до втрати доступності [1, 28].

Реагування на DDoS-активність може мати різні форми. Найпростішим варіантом є блокування IP-адреси, з якої надходить надмірна кількість пакетів. Іншим варіантом є обмеження частоти запитів, блокування певного типу трафіку, відхилення пакетів до конкретного порту або тимчасове посилення правил фільтрації. У контексті Raspberry Pi та Linux-середовища така реакція може реалізовуватися через iptables, nftables або інші механізми керування мережевими правилами. Важливим є те, що рішення про фільтрацію приймається до того, як надмірний трафік повністю навантажить IoT-пристрій [10, 42].

Для IoT-інфраструктури DDoS-атака небезпечна ще й тим, що вона може маскувати інші дії зловмисника. Поки система або адміністратор реагує на перевантаження, паралельно може виконуватися сканування портів, спроба підбору пароля, перевірка відкритих сервісів або пошук вразливого пристрою. У такій ситуації журналювання подій має не менше значення, ніж саме блокування трафіку. Запис часу атаки, IP-адрес джерел, типів пакетів, кількості звернень і застосованих правил дозволяє відновити картину події та оцінити, які вузли були найбільш уразливими [15, 45].

У бакалаврській роботі апаратно-програмний засіб на базі Raspberry Pi розглядається як проміжний захисний вузол, який не замінює провайдерський або хмарний DDoS-захист, але підвищує стійкість локальної IoT-мережі. Його завдання полягає у виявленні аномального трафіку на ранньому етапі, зменшенні навантаження на захищені пристрої, автоматичному застосуванні правил

фільтрації та збереженні інформації про події. Такий підхід є реалістичним для лабораторного стенда, оскільки не потребує складної мережевої інфраструктури та дозволяє показати повний цикл роботи: надходження трафіку, аналіз, виявлення аномалії, реагування і відображення результату [17, 39].

1.3 Ознаки аномального мережевого трафіку в IoT-сегменті

Аномальний мережевий трафік в IoT-сегменті можна розглядати як таку поведінку мережевих пакетів, яка помітно відрізняється від звичайного режиму роботи пристроїв. Для класичної комп'ютерної мережі тимчасове зростання навантаження не завжди є критичним, оскільки сервери, робочі станції та мережеве обладнання зазвичай мають певний запас продуктивності. У випадку IoT-середовища ситуація є складнішою, адже більшість пристроїв працює з обмеженими апаратними ресурсами, невеликим обсягом пам'яті, спрощеною мережевою логікою та невисокою здатністю до самостійної фільтрації підозрілих з'єднань. Через це навіть порівняно невелике відхилення від нормального профілю трафіку може спричинити помітне погіршення роботи пристрою, затримки, зависання або втрату доступності [9, 37].

У нормальному режимі IoT-пристрої зазвичай генерують передбачуваний трафік. Датчик температури або вологості надсилає коротке повідомлення через певний проміжок часу, контролер освітлення приймає команду лише після зміни стану, розумна розетка періодично передає інформацію про споживання електроенергії, а камера відеоспостереження формує більш інтенсивний, але все одно відносно стабільний потік даних. Такий характер роботи дозволяє сформувати уявлення про звичайну поведінку пристрою. Якщо в певний момент кількість пакетів, підключень або звернень до порту різко зростає без очевидної причини, це може вказувати на появу аномальної активності [4, 22].

Однією з основних ознак аномального трафіку є різке збільшення кількості пакетів за короткий проміжок часу. У стабільній IoT-мережі більшість пристроїв

не потребує постійного високочастотного обміну даними. Якщо за одну або кілька секунд кількість пакетів до певного вузла зростає в десятки разів порівняно зі звичайним режимом, фільтр має зафіксувати таке відхилення. Особливо важливо враховувати не лише загальний обсяг трафіку, а й те, до якого саме пристрою він спрямований. Для одного вузла кілька сотень пакетів за секунду можуть бути допустимими, а для простого сенсорного модуля така інтенсивність уже може стати критичною [15, 41].

У процесі виявлення аномального трафіку важливо відрізняти короткочасне збільшення активності від справді небезпечної ситуації. Не кожне перевищення середнього рівня означає DDoS-атаку. Наприклад, оновлення прошивки, перезапуск пристрою, підключення нового вузла, передавання відеопотоку або активне використання вебпанелі може тимчасово збільшити кількість пакетів. Через це фільтр має враховувати тривалість відхилення. Якщо висока активність триває лише короткий момент і швидко повертається до нормального рівня, її можна розглядати як допустиме коливання. Якщо ж перевищення повторюється або утримується протягом кількох часових вікон, така поведінка має більшу ймовірність бути аномальною [17, 40].

У межах цієї бакалаврської роботи ознаки аномального мережевого трафіку виступають основою для подальшої побудови апаратно-програмного засобу. Raspberry Pi як проміжний вузол має фіксувати параметри пакетів, порівнювати їх із допустимими межами, виявляти відхилення та передавати інформацію до модуля реагування. Саме на основі таких ознак система може приймати рішення про блокування IP-адреси, обмеження частоти запитів, фільтрацію певного типу пакетів або запис події до журналу. Це створює зв’язок між теоретичним аналізом трафіку та практичною реалізацією захисту IoT-пристроїв від DDoS-активності [19, 44].

1.4 Аналіз існуючих засобів захисту від DDoS-атак

Захист від DDoS-атак є одним із важливих напрямів забезпечення доступності комп'ютерних мереж, серверних систем та IoT-інфраструктури. На відміну від атак, спрямованих переважно на викрадення даних або несанкціонований доступ, DDoS-атаки порушують нормальну роботу системи через надмірне навантаження на мережевий канал, обчислювальні ресурси, таблиці з'єднань або прикладні сервіси. Через це засоби протидії таким атакам мають не лише блокувати шкідливі запити, а й підтримувати доступність легітимного трафіку, не створюючи значних затримок для звичайних користувачів і пристроїв. У контексті IoT-середовища це питання є особливо важливим, оскільки багато пристроїв мають обмежені ресурси та не можуть самостійно витримувати значне мережеве навантаження [6, 34].

Існуючі засоби захисту від DDoS-атак можна умовно поділити на кілька груп. До них належать хмарні сервіси фільтрації, провайдерські системи очищення трафіку, апаратні міжмережеві екрани, IDS/IPS-системи, WAF-рішення, програмні Linux-фільтри, механізми rate limiting, списки контролю доступу та локальні шлюзи безпеки. Кожен із цих підходів має власну сферу застосування, переваги та обмеження. Для великих компаній зазвичай використовуються хмарні або провайдерські рішення, які здатні обробляти дуже великі обсяги трафіку. Для малих мереж, навчальних стендів і локальних IoT-сегментів більш доцільними є програмно-апаратні рішення, які можна розгорнути без складної інфраструктури [11, 42].

Одним із найпотужніших підходів є використання хмарних сервісів захисту від DDoS-атак. Такі рішення працюють за принципом перенаправлення трафіку через інфраструктуру постачальника послуги, де виконується попередній аналіз, фільтрація та відкидання шкідливих пакетів. До таких засобів належать сервіси, орієнтовані на захист вебресурсів, API, DNS, корпоративних мереж і великих серверних інфраструктур. Їхня головна перевага полягає у здатності приймати та обробляти значні обсяги трафіку, які локальна мережа або окремий сервер не змогли б витримати самостійно. За рахунок розподіленої

інфраструктури хмарний постачальник може поглинати атаку ще до того, як вона досягне захищеного ресурсу [3, 27].

На рисунку 1.1 показано узагальнену класифікацію існуючих засобів захисту від DDoS-атак, які можуть застосовуватися в комп'ютерних мережах та IoT-інфраструктурі.

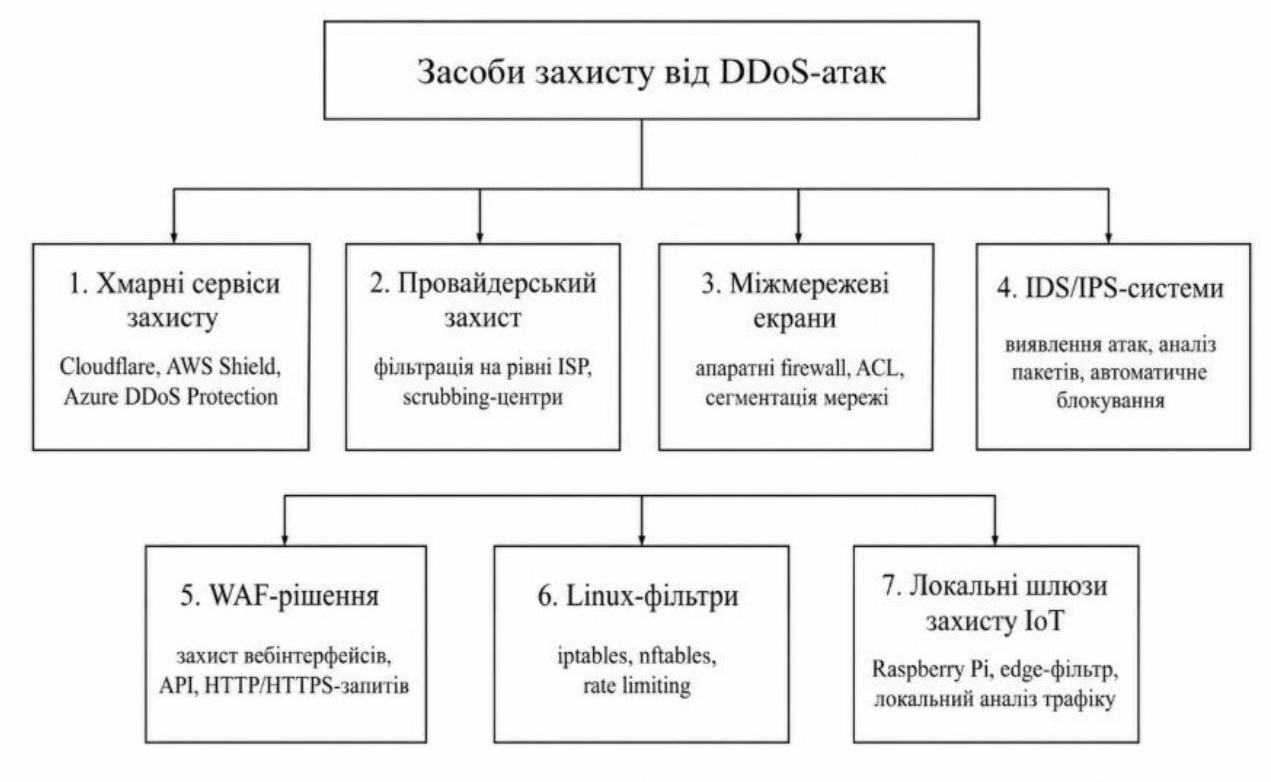


Рисунок 1.1 – Класифікація існуючих засобів захисту від DDoS-атак

Принцип роботи хмарного DDoS-захисту полягає у тому, що вхідний трафік спочатку потрапляє до захисної інфраструктури, де перевіряється за різними ознаками: джерело, частота запитів, тип протоколу, репутація IP-адреси, повторюваність пакетів, поведінка HTTP-запитів або характер DNS-звернень. Легітимний трафік пропускається до сервера або мережі користувача, а підозрілий блокується або обмежується. Такий підхід є ефективним для великих вебсервісів, інтернет-магазинів, банківських платформ, корпоративних порталів і публічних API. Проте для невеликої IoT-мережі він не завжди є зручним,

оскільки частина IoT-пристроїв працює локально, не має публічного доменного імені або не потребує складного хмарного маршруту [14, 39].

На рисунку 1.2 показано загальну схему роботи хмарного сервісу захисту, у якій трафік спочатку проходить через зовнішній центр фільтрації, а вже потім потрапляє до захищеної інфраструктури.

Рисунок 1.2 – Схема роботи хмарного сервісу захисту від DDoS-атак

Перевагою хмарних рішень є висока масштабованість. Якщо атака має великий обсяг, локальний маршрутизатор або сервер може бути перевантажений ще до того, як правила фільтрації почнуть працювати. Хмарний сервіс, навпаки, має значно більший запас пропускної здатності та може прийняти на себе основний потік шкідливого трафіку. Однак така модель має і певні обмеження. Вона залежить від зовнішнього постачальника, потребує правильного налаштування DNS, маршрутизації або тунелювання, може мати вартість, яка є надмірною для малої IoT-інфраструктури. Крім того, хмарний захист не завжди бачить внутрішній локальний трафік між пристроями, тому не вирішує всіх проблем захисту IoT-сегмента [8, 31].

На рисунку 1.3 показано приклад використання міжмережевого екрана для відокремлення IoT-сегмента від основної локальної мережі та зовнішнього трафіку.

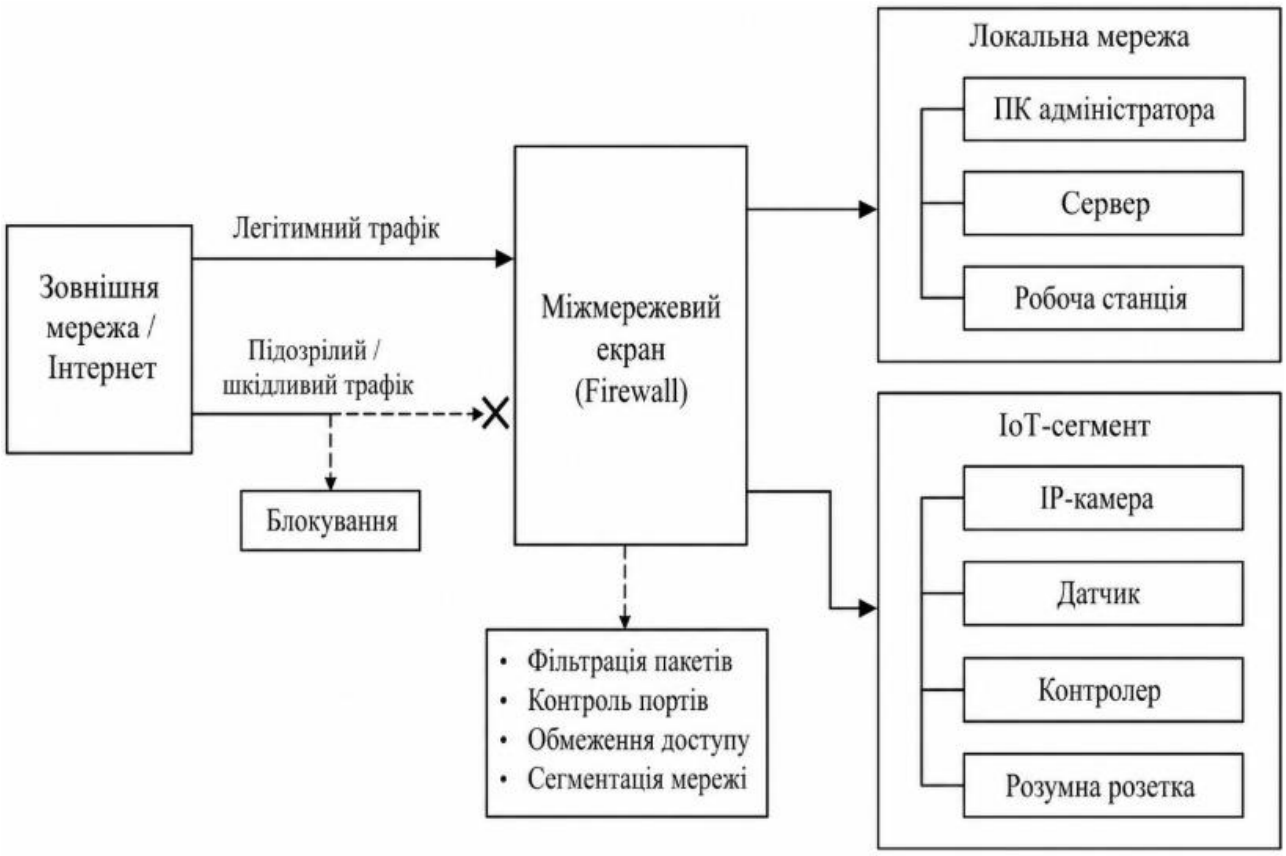


Рисунок 1.3 – Використання міжмережевого екрана для захисту IoT-сегмента

Окрему групу засобів становлять IDS та IPS-системи. IDS призначена для виявлення підозрілої активності, аналізу мережевого трафіку та формування сповіщень про можливі атаки. IPS виконує схожі функції, але додатково може активно втручатися в мережевий потік і блокувати небезпечні пакети. Такі системи можуть використовувати сигнатурний аналіз, евристичні правила, поведінкові моделі та статистичні показники. Вони корисні для виявлення сканування портів, спроб експлуатації вразливостей, підозрілих з'єднань і деяких видів DDoS-активності [4, 22].

IDS/IPS-рішення мають важливу перевагу: вони не обмежуються простим блокуванням за адресою або портом, а можуть аналізувати зміст пакетів, послідовність дій, повторюваність запитів і відповідність відомим шаблонам атак. Наприклад, система може виявляти велику кількість SYN-пакетів, аномальну UDP-активність, часті ICMP-запити або звернення до вразливих сервісів. Для IoT-середовища це корисно, оскільки багато пристроїв мають

типові слабкі місця, а атаки часто повторюють відомі сценарії. Водночас повноцінні IDS/IPS можуть вимагати значних ресурсів, правильного налаштування правил, регулярного оновлення сигнатур і досвіду адміністратора [13, 47].

Для захисту вебресурсів часто використовуються WAF-рішення, тобто міжмережеві екрани прикладного рівня. Їхнє призначення полягає у перевірці HTTP- та HTTPS-запитів, виявленні підозрілих параметрів, обмеженні небезпечних дій, блокуванні автоматизованих запитів і захисті вебзастосунків від типових атак. У контексті DDoS-захисту WAF може допомагати протидіяти атакам прикладного рівня, коли зловмисник не просто надсилає пакети, а багаторазово звертається до сторінки, API або вебінтерфейсу. Для IoT-шлюзів і пристроїв із вебпанелями такий підхід може бути корисним, оскільки багато систем керування мають саме вебінтерфейс [7, 33].

Однак WAF не є універсальним засобом захисту від усіх DDoS-атак. Він добре працює з прикладним трафіком, але не завжди допомагає проти низькорівневих атак на мережевий канал, TCP-стек, UDP-порти або ICMP-запити. Якщо IoT-пристрій перевантажується ще на рівні приймання пакетів, WAF може не встигнути обробити таку ситуацію або взагалі не брати участі в цьому потоці. Через це WAF доцільно розглядати як частину загальної системи захисту, а не як самостійне рішення для всього IoT-сегмента [18, 40].

Простішим, але практичним засобом є використання механізмів rate limiting, тобто обмеження частоти запитів. Такий підхід полягає в тому, що для певної IP-адреси, порту, протоколу або сервісу встановлюється допустима кількість запитів за одиницю часу. Якщо джерело перевищує цей поріг, його пакети тимчасово блокуються, сповільнюються або відкидаються. Для IoT-сегмента це дуже корисно, оскільки більшість пристроїв у нормальному режимі не потребує великої кількості звернень. Наприклад, якщо датчик або контролер отримує сотні запитів за секунду, така активність може вважатися підозрілою [5, 26].

На рисунку 1.4 показано місце WAF-рішення у захисті вебінтерфейсу IoT-шлюзу або сервера керування, де основна увага приділяється перевірці прикладних HTTP-запитів.

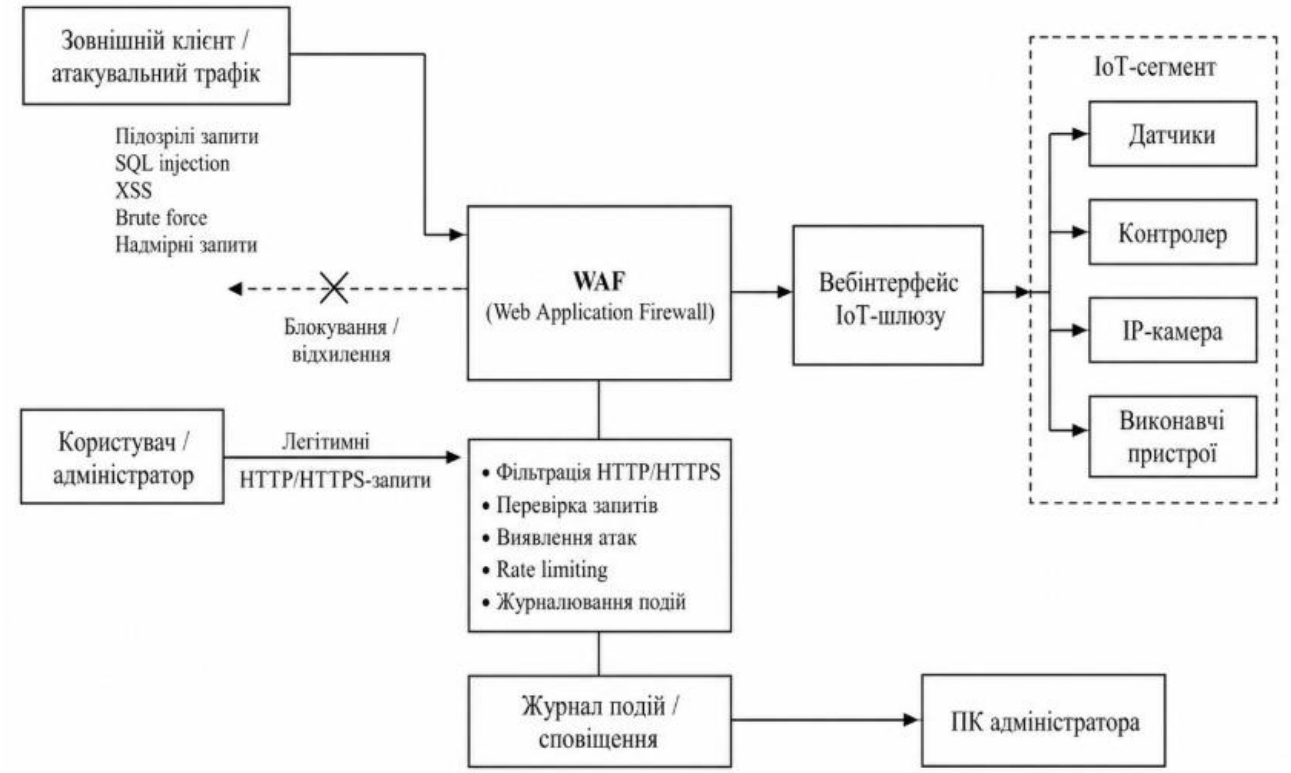


Рисунок 1.4 – Використання WAF для захисту вебінтерфейсу IoT-шлюзу

Перевага rate limiting полягає у відносній простоті реалізації. Його можна застосувати на маршрутизаторі, сервері, міжмережевому екрані або Linux-шлюзі. Такий механізм добре підходить для базового захисту від надмірної активності одного джерела, HTTP flood, частих підключень або невеликих UDP/ICMP-атак. Однак він має і недоліки. Якщо атака розподілена між великою кількістю джерел, кожне окреме джерело може не перевищувати встановлений поріг, але сумарне навантаження все одно буде небезпечним. Крім того, неправильно підібрані пороги можуть блокувати легітимних користувачів або, навпаки, пропускати шкідливий трафік [12, 38].

Ще одним підходом є використання списків контролю доступу. ACL-правила дозволяють визначити, які адреси, порти або протоколи мають право

взаємодіяти з певним пристроєм. Наприклад, для IoT-сегмента можна дозволити доступ лише з локальної мережі, заборонити вхідні підключення з Інтернету, закрити невикористані порти або дозволити MQTT-з'єднання тільки з визначеного брокера. Такий підхід зменшує площину атаки, оскільки зловмисник не може звертатися до сервісів, які заборонені правилами. Водночас ACL не завжди достатньо для виявлення аномалій, оскільки вони переважно працюють за статичними правилами [2, 44].

Програмні засоби фільтрації в Linux, зокрема iptables та nftables, мають особливе значення для цієї бакалаврської роботи. Вони дозволяють створювати правила пропускання, блокування, перенаправлення та обмеження трафіку на рівні операційної системи. За їх допомогою можна фільтрувати пакети за IP-адресами, портами, протоколами, станом з'єднання, частотою пакетів та іншими параметрами. На базі Raspberry Pi такі засоби дають змогу створити гнучкий локальний фільтр, який не лише використовує готові правила, а й може змінювати їх автоматично після виявлення аномальної активності [15, 43].

Перевага Linux-фільтрації полягає в тому, що вона поєднує простоту доступного обладнання з достатньо потужними мережевими можливостями. Raspberry Pi може виконувати роль проміжного шлюзу, через який проходить трафік до IoT-сегмента. Програмний модуль аналізу може рахувати пакети, визначати джерела, фіксувати тип протоколу, порівнювати активність із порогами та передавати команду на створення правила блокування. У такому варіанті система не просто статично забороняє певні адреси, а реагує на поведінку мережі. Саме ця властивість робить програмну Linux-фільтрацію доречною основою для практичної реалізації в межах роботи [9, 30].

На рисунку 1.5 показано схему локального Linux-фільтра на базі Raspberry Pi, який розміщується між маршрутизатором та IoT-сегментом і виконує аналіз трафіку перед його передаванням до пристроїв.

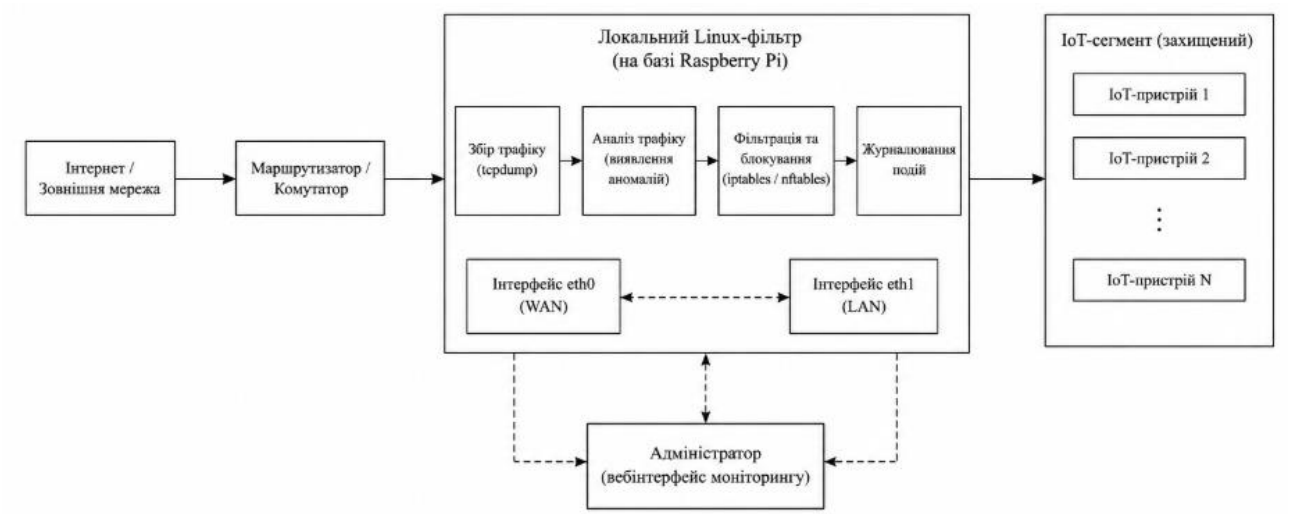


Рисунок 1.5 – Локальний Linux-фільтр на базі Raspberry Pi для захисту IoT-сегмента

Окрім класичних фільтрів, у мережевій безпеці можуть застосовуватися поведінкові та статистичні методи виявлення DDoS-активності. Вони ґрунтуються на порівнянні поточного трафіку з типовим профілем роботи системи. Для IoT-мереж такий підхід є досить зручним, оскільки багато пристроїв мають стабільну частоту обміну даними. Якщо сенсор зазвичай передає коротке повідомлення раз на кілька секунд, а потім раптово отримує велику кількість пакетів, така зміна може бути зафіксована як аномалія. На відміну від сигнатурного підходу, поведінковий аналіз може виявляти не лише відомі атаки, а й нестандартні відхилення від нормальної роботи [16, 41].

З огляду на особливості теми роботи найбільш доцільним є підхід, у якому Raspberry Pi виконує роль локального апаратно-програмного фільтра. Такий засіб не претендує на заміну великих хмарних або провайдерських систем, але дозволяє вирішити практичну задачу захисту невеликого IoT-сегмента. Його перевага полягає в тому, що він може бути розгорнутий без дорогого обладнання, підтримує стандартні Linux-інструменти, дозволяє реалізувати власну логіку аналізу та забезпечує наочну перевірку результатів у лабораторних умовах. Для бакалаврської роботи це є суттєвою перевагою, оскільки система має бути не лише теоретично описана, а й практично реалізована [24, 47].

У межах цієї роботи існуючі засоби захисту від DDoS-атак розглянуто як основу для формування власного рішення. Хмарні та провайдерські сервіси демонструють важливість попередньої фільтрації трафіку до того, як він досягне цільового вузла. IDS/IPS-системи показують значення аналізу поведінки та виявлення підозрілих ознак. Міжмережеві екрани й ACL-правила підтверджують потребу в обмеженні доступу до непотрібних портів і сервісів. Rate limiting підкреслює важливість контролю частоти запитів. Linux-фільтрація надає технічну основу для практичної реалізації такого підходу на Raspberry Pi [25, 32].

1.5 Постановка задачі

У попередніх підрозділах розглянуто особливості побудови IoT-мереж, загальну характеристику DDoS-атак, ознаки аномального мережевого трафіку та існуючі засоби захисту від перевантаження мережевої інфраструктури. Проведений аналіз показав, що IoT-сегмент має низку специфічних особливостей, які відрізняють його від класичних комп'ютерних мереж. До таких особливостей належать обмежені обчислювальні ресурси пристроїв, різнорідність апаратного забезпечення, використання спрощених прошивок, постійне підключення до мережі, обмежені можливості самостійного захисту та залежність від стабільного обміну даними. Через це навіть незначне зростання кількості пакетів або підключень може негативно вплинути на доступність пристрою, затримати передавання телеметрії, порушити роботу вебінтерфейсу або спричинити втрату зв'язку з керуючим вузлом [6, 31].

У межах бакалаврської роботи сформульовано задачу розроблення апаратно-програмного засобу для захисту IoT-пристроїв від DDoS-атак шляхом створення локального фільтра мережевого трафіку на базі Raspberry Pi. Такий засіб має виконувати роль проміжного вузла між основною мережею та захищеним IoT-сегментом. Його призначення полягає у спостереженні за мережевим трафіком, визначенні ознак аномальної активності, фіксації

підозрілих джерел, застосуванні правил фільтрації та збереженні інформації про події безпеки.

Актуальність поставленої задачі пов’язана з тим, що багато IoT-пристроїв не мають власних ефективних механізмів протидії мережевим атакам. Звичайний датчик, камера, контролер або мікроконтролерний вузол здебільшого розрахований на виконання конкретної функції: зчитування даних, передавання повідомлень, приймання команд або керування виконавчим елементом. Такі пристрої не завжди можуть аналізувати потік пакетів, рахувати кількість підключень, відрізняти нормальний трафік від аномального або автоматично блокувати підозрілі IP-адреси.

Об’єктом бакалаврської роботи є процес захисту IoT-пристроїв від аномального мережевого навантаження в локальній мережі. У цьому процесі основна увага зосереджена на збереженні доступності пристроїв, зменшенні впливу надмірного трафіку та своєчасному виявленні ознак DDoS-активності. Предметом бакалаврської роботи є апаратно-програмний засіб на базі Raspberry Pi, який виконує аналіз мережевого трафіку, виявлення аномалій, фільтрацію підозрілих пакетів і журналювання подій.

Метою бакалаврської роботи є розроблення апаратно-програмного засобу для захисту IoT-пристроїв від DDoS-атак, який на базі Raspberry Pi забезпечує виявлення аномального мережевого трафіку, автоматичну фільтрацію підозрілих з’єднань і відображення подій безпеки. Досягнення цієї мети передбачає поєднання апаратної платформи, мережевих механізмів Linux, програмного модуля аналізу трафіку, алгоритму визначення аномальної активності та простого інтерфейсу моніторингу. У такій побудові Raspberry Pi виконує роль доступного, гнучкого й достатньо продуктивного вузла, який може бути використаний у лабораторному стенді або невеликому локальному середовищі [18, 45].

1.6 Висновки до першого розділу

У першому розділі бакалаврської роботи розглянуто проблему захисту IoT-пристроїв від DDoS-атак та визначено основні чинники, які роблять такі пристрої вразливими до аномального мережевого навантаження. Проаналізовано особливості побудови та функціонування IoT-мереж, у яких одночасно можуть працювати датчики, контролери, IP-камери, шлюзи, мікроконтролерні вузли та інші пристрої з різним рівнем продуктивності й різними можливостями захисту. Показано, що IoT-сегмент відрізняється від звичайної локальної мережі тим, що значна частина його елементів має обмежені апаратні ресурси, спрощену програмну логіку, мінімальні засоби самостійного аналізу трафіку та часто працює у безперервному режимі.

У розділі охарактеризовано DDoS-атаки як один із найбільш небезпечних видів мережевого впливу на IoT-інфраструктуру. Визначено, що головна мета таких атак полягає не стільки в отриманні доступу до даних, скільки в порушенні доступності пристроїв, сервісів або цілих мережевих сегментів. Для IoT-пристроїв така загроза є особливо відчутною, оскільки навіть помірне збільшення кількості пакетів може спричинити затримки, зависання, розрив з’єднання або втрату працездатності вузла.

Також проаналізовано існуючі засоби захисту від DDoS-атак, зокрема хмарні сервіси, провайдерську фільтрацію, міжмережеві екрани, IDS/IPS-системи, WAF-рішення, механізми rate limiting, ACL-правила та Linux-фільтри. Встановлено, що потужні комерційні та хмарні рішення є ефективними для великих інфраструктур, однак для невеликого локального IoT-сегмента вони не завжди є зручними, доступними або виправданими за складністю впровадження. У результаті обґрунтовано потребу у створенні локального апаратно-програмного засобу, який працює безпосередньо перед IoT-сегментом, аналізує трафік і реагує на підозрілу активність до того, як вона створить критичне навантаження на захищені пристрої.

2 РОЗРОБЛЕННЯ АПАРАТНО-ПРОГРАМНОГО ЗАСОБУ ЗАХИСТУ ІОТ-ПРИСТРОЇВ

2.1 Загальна архітектура апаратно-програмного засобу захисту IoT-пристроїв

У другому розділі бакалаврської роботи розглянуто побудову апаратно-програмного засобу, призначеного для захисту IoT-пристроїв від DDoS-атак у локальному мережевому середовищі. Після аналізу особливостей IoT-мереж, характеру DDoS-атак та ознак аномального трафіку сформовано загальне бачення системи, у якій захисна логіка не покладається безпосередньо на малоресурсні пристрої, а виноситься на окремий проміжний вузол. Такий підхід є доцільним, оскільки більшість IoT-пристроїв не має достатньої продуктивності для самостійного аналізу великої кількості пакетів, ведення журналу подій, блокування підозрілих адрес і динамічної зміни правил фільтрації.

Основою запропонованого апаратно-програмного засобу є Raspberry Pi, який виконує роль локального фільтра між основною мережею та захищеним IoT-сегментом. У такій архітектурі Raspberry Pi не просто підключається до мережі як звичайний пристрій, а працює як контрольна точка, через яку проходить або аналізується трафік, спрямований до IoT-вузлів. Це дозволяє перевіряти мережеву активність до того моменту, коли вона потрапляє безпосередньо на захищений пристрій. Саме таке розміщення має важливе значення, адже під час DDoS-активності головним завданням є зменшення навантаження на слабкий вузол ще до його фактичного перевантаження.

Загальна ідея архітектури полягає в тому, що IoT-пристрої не приймають увесь вхідний трафік напряму з основної мережі. Перед ними розміщено Raspberry Pi, на якому працює програмна частина засобу захисту. Вона аналізує параметри пакетів, визначає інтенсивність звернень, контролює IP-адреси джерел, типи протоколів, порти призначення та частоту надходження пакетів. Якщо активність відповідає нормальному режиму, трафік пропускається до IoT-

сегмента. Якщо ж система виявляє різке зростання кількості пакетів, велику кількість однотипних запитів, підозрілу активність з однієї адреси або тривале перевищення встановлених меж, формується подія безпеки та активується механізм реагування.

У межах бакалаврської роботи архітектуру засобу побудовано як поєднання апаратної та програмної частини. Апаратна частина охоплює Raspberry Pi, мережеве підключення, маршрутизатор або комутатор, захищені IoT-пристрої та комп'ютер адміністратора. Програмна частина складається з модулів зчитування трафіку, попередньої обробки, виявлення аномалій, фільтрації, журналювання та відображення стану системи. Такий поділ дозволяє представити засіб не як один окремий скрипт або набір команд, а як цілісну систему з чітко визначеними функціональними блоками.

У запропонованій архітектурі основна мережа виступає середовищем, з якого може надходити як звичайний, так і потенційно небезпечний трафік. До цієї частини можна віднести маршрутизатор, комп'ютер адміністратора, зовнішній сервер, тестовий вузол або пристрій, який генерує навантаження під час перевірки працездатності засобу. З іншого боку розміщується IoT-сегмент, до якого входять пристрої, що потребують захисту. Це можуть бути сенсори, контролери, IP-камери, мікроконтролерні вузли, локальний MQTT-брокер, вебсервер пристрою або програмний емулятор IoT-вузла. Між цими двома частинами встановлено Raspberry Pi, який бере на себе роль проміжного фільтрувального вузла. Роль Raspberry Pi в цій архітектурі є центральною. Він виконує не лише функцію фізичного посередника між мережевими сегментами, а й функцію програмного аналізатора.

На рисунку 2.1 показано загальну архітектуру апаратно-програмного засобу захисту IoT-пристроїв від DDoS-атак, у якій Raspberry Pi розміщено між основною мережею та захищеним IoT-сегментом.

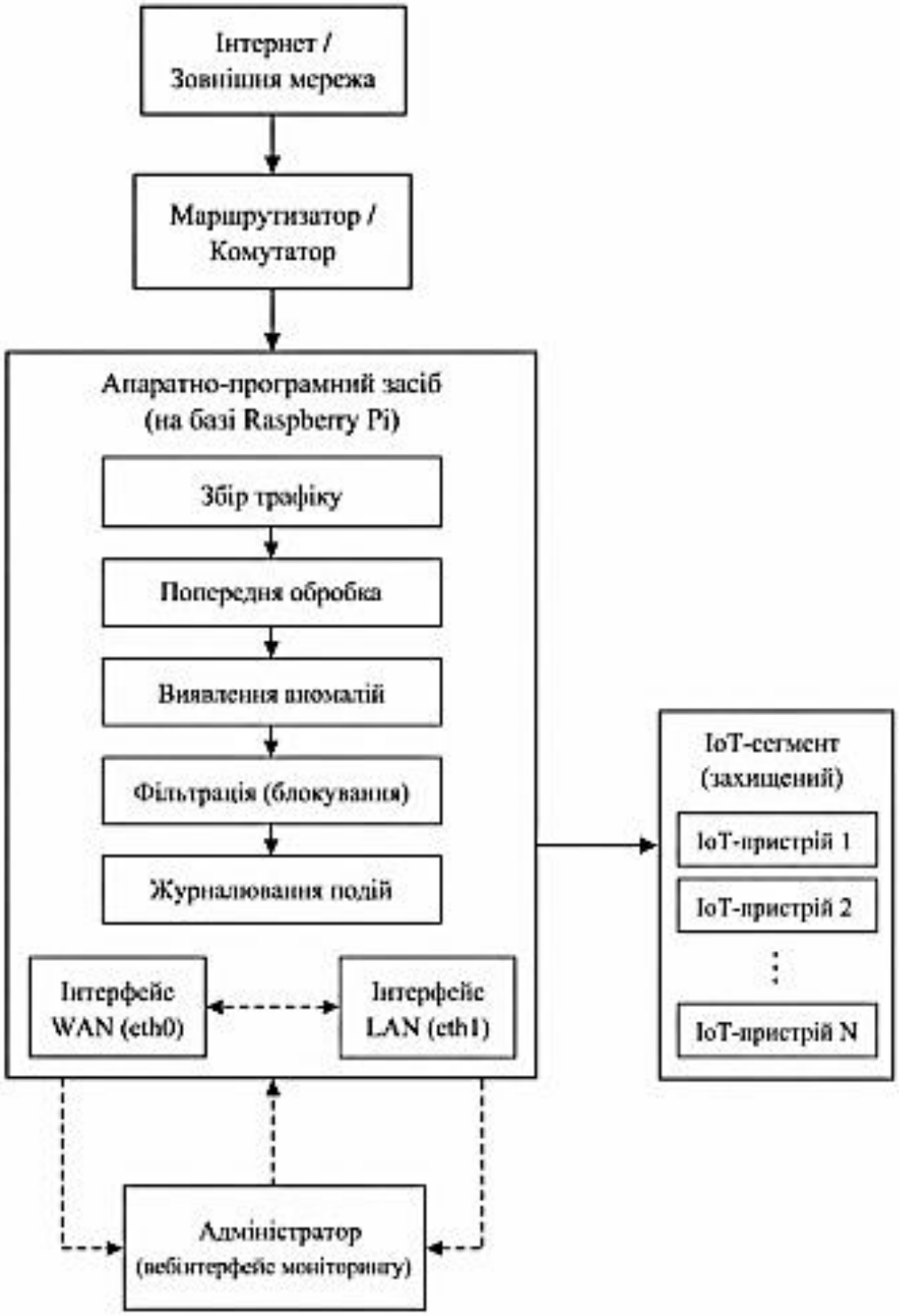


Рисунок 2.1 – Загальна архітектура апаратно-програмного засобу захисту IoT-пристроїв від DDoS-атак

На ньому розміщено основну логіку засобу захисту: приймання даних про трафік, підрахунок мережевих показників, перевірку на аномальність, створення

правил блокування та збереження інформації про події. Завдяки підтримці Linux-середовища Raspberry Pi дозволяє використовувати стандартні механізми роботи з мережею, зокрема системні засоби фільтрації пакетів, локальні журнали, скрипти автоматизації та вебсервер для відображення стану системи.

Першим логічним модулем архітектури є модуль зчитування мережевого трафіку. Його завдання полягає в отриманні базових параметрів пакетів, які надходять до захищеного сегмента або проходять через фільтр. Для роботи засобу не обов'язково зберігати повний вміст кожного пакета, оскільки основну цінність мають службові характеристики: IP-адреса джерела, IP-адреса призначення, протокол, порт, розмір пакета, напрям руху та час надходження. Такий підхід зменшує навантаження на Raspberry Pi та дозволяє зосередитися саме на тих ознаках, які є важливими для виявлення DDoS-активності.

Після отримання параметрів пакетів дані передаються до модуля попередньої обробки. У цьому модулі трафік групується за часовими інтервалами, джерелами, цільовими адресами, протоколами та портами. Наприклад, система може підраховувати кількість пакетів за одну секунду, кількість TCP SYN-пакетів за короткий проміжок часу, кількість UDP-пакетів до певного порту або кількість ICMP-запитів до конкретного пристрою. Така попередня обробка потрібна для того, щоб система працювала не з окремими випадковими пакетами, а з узагальненими показниками мережевої активності.

Архітектура засобу передбачає декілька режимів роботи. У нормальному режимі трафік відповідає встановленим межам, тому система лише спостерігає за пакетами, оновлює статистику та зберігає службові дані. У режимі підозрілої активності окремі показники наближаються до порогів або незначно їх перевищують, через що система формує попередження та посилює контроль за певним джерелом або портом. У режимі активної атаки перевищення має стійкий характер, тому запускається механізм реагування та створюються правила фільтрації. Такий поділ дозволяє зробити роботу засобу більш обережною й не блокувати трафік одразу після кожного незначного відхилення.

Важливою перевагою запропонованої архітектури є модульність. Кожен компонент має власне призначення і може розроблятися окремо: модуль зчитування трафіку відповідає за отримання даних, модуль обробки формує показники, модуль виявлення аномалій приймає рішення, модуль реагування застосовує захисні дії, журнал зберігає інформацію, а вебінтерфейс відображає стан системи. Завдяки такому підходу засіб можна поступово розширювати. Наприклад, у майбутньому до нього може бути додано складніший алгоритм аналізу, підтримку нових типів атак, інтеграцію з віддаленим сервером моніторингу або систему сповіщень.

2.2 Обґрунтування вибору апаратних і програмних компонентів

Для реалізації апаратно-програмного засобу захисту IoT-пристроїв від DDoS-атак важливо обрати такі компоненти, які відповідають одразу кільком вимогам: доступність, достатня продуктивність, простота налаштування, підтримка мережевих функцій, можливість програмного розширення та придатність до використання в лабораторному стенді. Оскільки бакалаврська робота орієнтована не на створення промислового центру очищення трафіку, а на побудову локального фільтра для IoT-сегмента, апаратна частина має залишатися компактною та зрозумілою. Водночас вона повинна забезпечувати реальну можливість аналізу мережевого трафіку, фільтрації пакетів, журналювання подій і відображення результатів роботи системи.

Основною апаратною платформою обрано Raspberry Pi. Такий вибір пояснюється тим, що ця одноплатна комп’ютерна система поєднує невеликі фізичні розміри, достатню продуктивність для оброблення локального мережевого трафіку, підтримку операційних систем на основі Linux і можливість підключення додаткових мережевих адаптерів. Raspberry Pi є зручною платформою для навчальних і прикладних проєктів, оскільки на ній можна запускати повноцінні програмні модулі, працювати з мережевими інтерфейсами,

налаштовувати фільтрацію пакетів і створювати вебінтерфейс для моніторингу. У межах цієї роботи Raspberry Pi виконує роль центрального вузла, через який проходить або контролюється трафік між основною мережею та захищеним IoT-сегментом.

Вибір Raspberry Pi також обґрунтовано тим, що платформа не обмежується функціями мікроконтролера. На відміну від простих плат на базі мікроконтролерів, Raspberry Pi має повноцінне програмне середовище, файлову систему, підтримку багатозадачності, системні журнали, мережеві служби та можливість запуску скриптів у фоновому режимі. Це дає змогу реалізувати засіб захисту не як простий пристрій із фіксованою логікою, а як гнучку систему, у якій окремі функції виконують різні програмні модулі. Для задачі захисту від DDoS-активності це має важливе значення, оскільки потрібно не лише приймати трафік, а й аналізувати його поведінку в часі, зберігати події та змінювати правила фільтрації.

У запропонованій архітектурі Raspberry Pi може використовуватися як проміжний шлюз між двома мережевими зонами. Для цього доцільно застосувати вбудований Ethernet-інтерфейс і додатковий USB–Ethernet-адаптер. Один інтерфейс підключається до основної мережі або маршрутизатора, а другий - до захищеного IoT-сегмента. Такий поділ дозволяє фізично розмежувати напрям руху трафіку та зробити Raspberry Pi реальною контрольною точкою. Якщо використання другого мережевого адаптера неможливе, контроль може бути організовано логічно, але для практичної частини більш наочним є саме варіант із двома інтерфейсами. У цьому випадку схема роботи засобу стає зрозумілішою: трафік спочатку надходить на Raspberry Pi, проходить аналіз і лише після цього передається до IoT-пристроїв.

До складу експериментального стенда також входять маршрутизатор або комутатор, IoT-пристрій або його програмний емулятор, комп’ютер адміністратора та окремий вузол для генерації тестового навантаження. Маршрутизатор забезпечує підключення основної мережі, комутатор може

використовуватися для об’єднання декількох пристроїв у локальному сегменті, а комп’ютер адміністратора потрібний для налаштування Raspberry Pi, запуску тестів і перегляду вебінтерфейсу. Захищений IoT-пристрій може бути представлений реальним сенсорним або мікроконтролерним вузлом, невеликим вебсервером, MQTT-клієнтом або програмним емулятором, який імітує поведінку IoT-пристрою. Такий підхід дозволяє перевірити роботу засобу без потреби у великій кількості реального обладнання.

Програмну основу засобу становить операційна система Raspberry Pi OS або інший Linux-дистрибутив, придатний для роботи на Raspberry Pi. Вибір Linux-середовища є важливим, оскільки саме воно забезпечує доступ до мережевих інструментів, системних служб, механізмів маршрутизації та засобів фільтрації пакетів. У межах цієї роботи операційна система виконує роль базового середовища, на якому запускаються всі програмні модулі: аналізатор трафіку, модуль виявлення аномалій, механізм взаємодії з правилами фільтрації, журналювання та вебінтерфейс. Завдяки цьому апаратно-програмний засіб не потребує складного спеціалізованого програмного забезпечення, а спирається на стандартні можливості Linux.

Для відображення стану системи обрано вебінтерфейс. Його можна реалізувати за допомогою Flask або FastAPI, оскільки ці інструменти дозволяють створити просту локальну панель моніторингу без надмірного ускладнення програмної частини. Через вебінтерфейс може відображатися поточний стан фільтра, кількість оброблених пакетів, список активних підозрілих адрес, журнал подій, статус правил блокування та загальна інтенсивність трафіку. Такий підхід є зручним для демонстрації результатів, оскільки робота засобу стає видимою не лише через консоль, а й через зрозумілу графічну або табличну форму.

У структурі апаратних компонентів центральне місце займає Raspberry Pi, до якого підключаються мережеві інтерфейси, основна мережа та захищений IoT-сегмент. Додатковий USB–Ethernet-адаптер дозволяє розділити вхідний і вихідний напрям трафіку, що робить схему ближчою до реального мережевого

шлюзу. Маршрутизатор забезпечує зв’язок із основною мережею, а IoT-пристрій або його емулятор виступає цільовим вузлом, доступність якого потрібно захистити. Комп’ютер адміністратора використовується для налаштування, керування та перегляду результатів роботи системи. Такий склад є достатнім для лабораторної перевірки й не потребує дорогого обладнання.

На рисунку 2.2 показано структуру апаратних і програмних компонентів засобу захисту, де Raspberry Pi поєднує фізичну мережеву взаємодію з програмними модулями аналізу, фільтрації, журналювання та моніторингу.

Програмні компоненти взаємодіють між собою послідовно. Спочатку модуль зчитування отримує дані про пакети. Далі модуль попередньої обробки групує їх за часовими інтервалами, IP-адресами, портами та протоколами. Після цього модуль виявлення аномалій порівнює поточні показники з установленими порогами. Якщо виявлено перевищення, модуль реагування формує команду для зміни правил фільтрації. Паралельно подія записується до локальної бази даних, а вебінтерфейс отримує оновлену інформацію для відображення. Така послідовність забезпечує повний цикл роботи засобу від приймання трафіку до фіксації результату.

Перевагою обраного набору компонентів є його узгодженість із метою бакалаврської роботи. Raspberry Pi забезпечує апаратну основу, Linux надає мережеві можливості, Python дозволяє реалізувати логіку аналізу, iptables або nftables виконують реальну фільтрацію пакетів, SQLite зберігає події, а вебінтерфейс забезпечує наочне відображення роботи системи. Усі компоненти взаємно доповнюють одне одного, тому засіб не виглядає як випадковий набір програм, а формується як цілісна система з чітким розподілом функцій.

Ще однією перевагою такого підходу є можливість поступового нарощування функціональності. На першому етапі система може лише рахувати пакети та відображати статистику. На наступному етапі додається порогове виявлення аномалій. Далі реалізується автоматичне блокування підозрілих джерел, журналювання та вебпанель. У майбутньому до такої системи можна

додати сповіщення, складніший аналіз трафіку, підтримку кількох IoT-сегментів або інтеграцію з віддаленим сервером моніторингу. У межах цієї роботи достатньо реалізувати базовий, але завершений набір функцій, який демонструє працездатність запропонованого рішення.

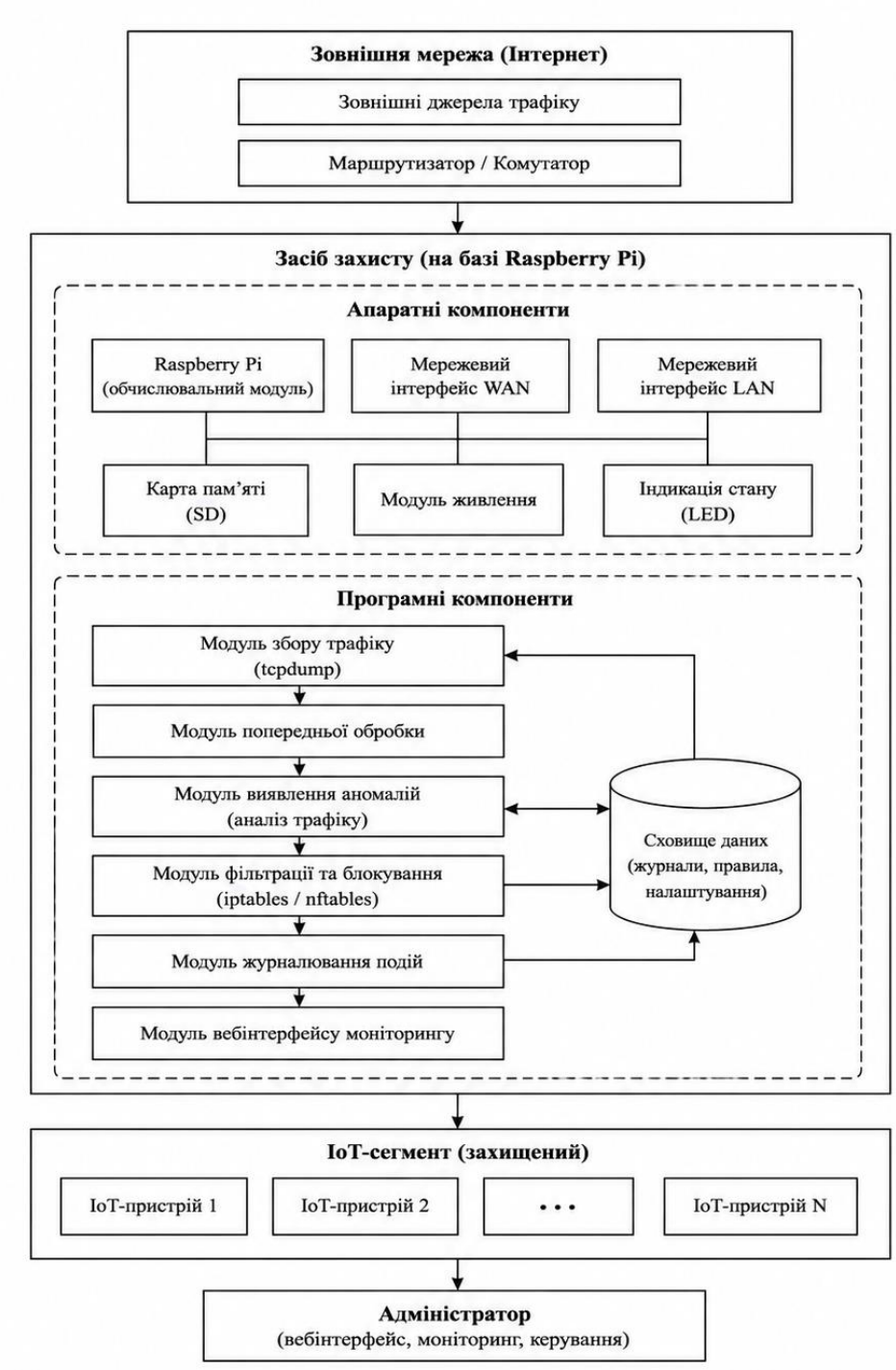


Рисунок 2.2 – Структура апаратних і програмних компонентів засобу захисту IoT-пристроїв

2.3 Розроблення структури мережевого фільтра на базі Raspberry Pi

Мережевий фільтр на базі Raspberry Pi у межах бакалаврської роботи розроблено як центральний вузол апаратно-програмного засобу захисту IoT-пристроїв від DDoS-атак. Його основне призначення полягає у розміщенні між основною мережею та захищеним IoT-сегментом, прийманні або спостереженні за мережевим трафіком, визначенні його характеристик, виявленні підозрілої активності та застосуванні правил фільтрації. На відміну від звичайного маршрутизатора, який переважно передає пакети між мережами відповідно до налаштувань маршрутизації, запропонований фільтр виконує додаткову аналітичну функцію. Він не лише пропускає або блокує трафік, а й оцінює його поведінку за певними ознаками.

Структуру мережевого фільтра сформовано з урахуванням того, що IoT-пристрої мають обмежені ресурси та не завжди здатні самостійно протидіяти надмірному потоку пакетів. Через це Raspberry Pi виступає окремим захисним вузлом, який бере на себе функції аналізу, контролю та реагування. Такий підхід дозволяє не змінювати програмне забезпечення самих IoT-пристроїв, а забезпечити їхній захист на рівні мережевої взаємодії. Це особливо важливо для готових датчиків, камер, контролерів або емуляторів IoT-вузлів, у яких немає можливості глибоко змінювати прошивку або додавати власні механізми кіберзахисту.

Окремий блок структури відповідає за прийняття рішення. Після того як модуль аналізу визначив підозрілу активність, система повинна обрати дію. У найпростішому випадку формується запис у журналі без негайного блокування. Такий варіант може застосовуватися для незначних перевищень або початкового режиму спостереження. Якщо ж активність має стійкий характер і повторюється протягом кількох часових інтервалів, фільтр може перейти до активного реагування. У цьому випадку створюється правило для обмеження або

блокування трафіку. Такий поділ потрібний для того, щоб система не реагувала надто жорстко на кожне короткочасне відхилення.

Блок фільтрації реалізує безпосередній вплив на мережевий потік. У структурі фільтра він взаємодіє із системними засобами Linux, які відповідають за оброблення пакетів. За допомогою правил iptables або nftables можна блокувати трафік від певної IP-адреси, обмежувати частоту пакетів, забороняти звернення до окремого порту або фільтрувати певний тип протоколу. Наприклад, якщо виявлено велику кількість TCP SYN-пакетів від одного джерела, система може створити тимчасове правило блокування цієї адреси. Якщо помічено надмірний ICMP-трафік, фільтр може обмежити частоту таких запитів. Якщо зафіксовано потік UDP-пакетів до невикористаного порту, відповідний трафік може бути відкинуто.

У структурі фільтра важливе місце займає блок журналювання. Він забезпечує збереження інформації про всі суттєві події, що відбуваються під час роботи системи. До журналу можуть потрапляти записи про перевищення порогів, виявлення підозрілих IP-адрес, створення правил блокування, завершення дії тимчасових правил, помилки оброблення трафіку та зміну режиму роботи системи. Кожен запис має містити час події, адресу джерела, адресу цільового пристрою, тип протоколу, кількість пакетів, причину спрацювання та виконану дію. Наявність такого журналу дозволяє надалі оцінити, як саме фільтр реагував на різні типи навантаження.

Для зберігання журналу може використовуватися локальна база даних SQLite або структурований файл. У межах прототипу більш зручним є використання SQLite, оскільки вона не потребує окремого серверного компонента та дозволяє швидко отримувати записи для відображення у вебінтерфейсі. У базі можуть зберігатися не лише події, а й поточні налаштування порогів, список заблокованих адрес, час початку та завершення блокування, а також службові дані про стан системи. Це робить мережевий

фільтр не просто тимчасовим скриптом, а більш завершеним програмним засобом із власною історією роботи.

Ще одним структурним елементом є вебінтерфейс моніторингу. Він потрібний для того, щоб адміністратор міг бачити стан фільтра без постійного перегляду консольних повідомлень. Через вебінтерфейс може відображатися поточна інтенсивність трафіку, кількість оброблених пакетів, перелік активних підозрілих джерел, список заблокованих IP-адрес, останні події журналу та поточний режим роботи системи. Для бакалаврської роботи такий компонент має велике значення, оскільки він робить практичну реалізацію більш наочною. Робота фільтра стає зрозумілою не лише на рівні коду, а й через візуальне відображення результатів.

Фільтр також передбачає роботу з різними режимами. У режимі спостереження система лише збирає статистику та не втручається у проходження пакетів. Такий режим корисний на етапі первинного налаштування, коли потрібно зрозуміти нормальний рівень трафіку для конкретної IoT-мережі. У режимі попередження система вже фіксує підозрілі події та відображає їх у журналі, але не завжди блокує джерело. У режимі активного захисту фільтр автоматично створює правила для обмеження або блокування трафіку. Така поетапність дозволяє налаштовувати систему поступово та уникати небажаного блокування легітимних з’єднань.

Для роботи мережевого фільтра важливо правильно визначити напрям руху трафіку. Пакети, які надходять із зовнішнього або основного сегмента до IoT-пристроїв, мають проходити основну перевірку. Вихідний трафік від IoT-пристроїв також може контролюватися, оскільки заражений пристрій потенційно може сам стати джерелом небажаної активності. Проте в межах цієї бакалаврської роботи головний акцент зроблено саме на захисті IoT-пристроїв від вхідного аномального навантаження. Це дозволяє чітко обмежити задачу й не перетворювати прототип на надмірно складну систему повного мережевого моніторингу.

На рисунку 2.3 показано структуру мережевого фільтра на базі Raspberry Pi, у якій виділено апаратні з’єднання, модулі програмної обробки трафіку, механізм фільтрації, журнал подій і вебінтерфейс моніторингу.

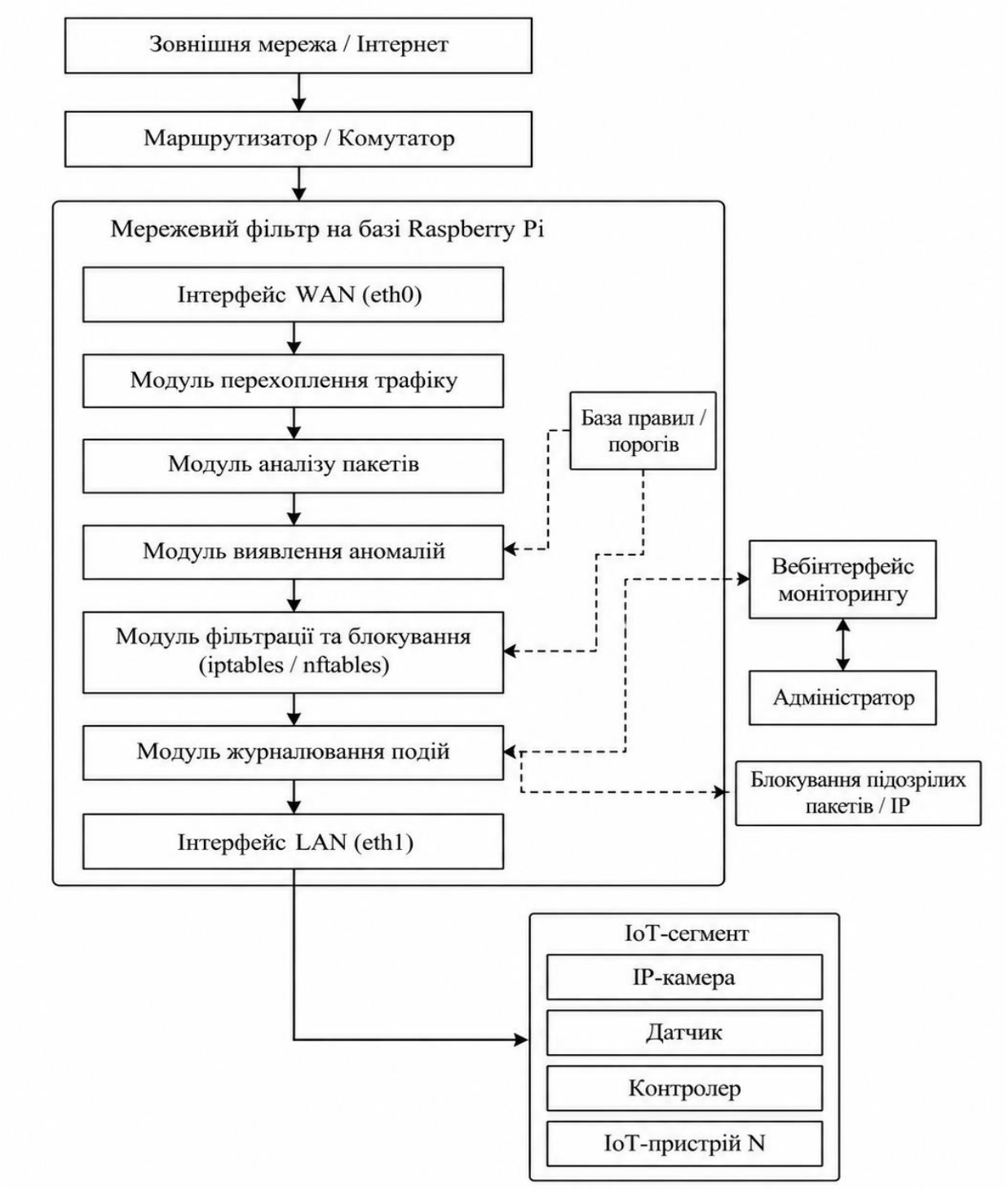


Рисунок 2.3 – Структура мережевого фільтра на базі Raspberry Pi

У цій структурі трафік рухається від основної мережі до Raspberry Pi, де проходить первинне зчитування та попереднє оброблення. Далі сформовані показники передаються до модуля аналізу, який визначає, чи відповідає активність нормальному режиму. Якщо трафік не має ознак аномальності, він пропускається до IoT-сегмента. Якщо виявлено перевищення встановлених меж, система формує подію, записує її до журналу та передає команду до блоку фільтрації. Після цього небажаний трафік може бути заблокований або обмежений, а адміністратор бачить відповідну інформацію у вебінтерфейсі.

Структура фільтра враховує потребу в гнучкому налаштуванні порогів. Для різних IoT-пристроїв нормальна інтенсивність трафіку може відрізнятися. Наприклад, простий датчик температури має передавати або приймати дуже мало пакетів, тоді як IP-камера може створювати значно більший потік даних. Через це в системі має бути передбачена можливість змінювати пороги для різних пристроїв або типів трафіку. Такий підхід дозволяє адаптувати фільтр до конкретного стенда, а не використовувати однакові правила для всіх вузлів без урахування їхньої поведінки.

Під час розроблення структури також враховано потребу в тимчасовому блокуванні. Постійне блокування IP-адреси після одного спрацювання може бути надто жорстким рішенням, особливо якщо перевищення виникло через випадкову або короткочасну активність. Тому доцільно передбачити блокування на певний період, після завершення якого правило може бути автоматично видалено. Якщо підозріла активність повторюється, система може знову застосувати блокування або збільшити його тривалість. Такий підхід робить захист більш гнучким і зменшує ризик довготривалого обмеження легітимного трафіку.

У межах запропонованої структури Raspberry Pi виконує одразу кілька ролей. На фізичному рівні він є проміжним вузлом між мережевими сегментами. На системному рівні він працює як Linux-шлюз і фільтр пакетів. На програмному рівні він виконує аналіз трафіку, ведення статистики, виявлення аномалій і

взаємодію з базою даних. На рівні користувацької взаємодії він надає вебінтерфейс для перегляду стану системи. Поєднання цих ролей дозволяє створити компактний, але функціонально завершений засіб захисту IoT-сегмента.

2.4 Розроблення алгоритму виявлення аномального мережевого трафіку

Алгоритм виявлення аномального мережевого трафіку є центральною частиною апаратно-програмного засобу захисту IoT-пристроїв від DDoS-атак. Саме від його логіки залежить, чи зможе система своєчасно відрізнити звичайний мережевий обмін від підозрілої активності, яка створює надмірне навантаження на захищені вузли. У межах бакалаврської роботи алгоритм побудовано з урахуванням особливостей IoT-сегмента, де трафік зазвичай має відносно передбачуваний характер, а самі пристрої не мають значного запасу продуктивності. Через це головну увагу приділено не складному глибокому аналізу вмісту пакетів, а контролю кількісних і поведінкових ознак мережевої активності.

Наступним етапом є формування статистичних показників за часове вікно. У межах одного вікна система визначає загальну кількість пакетів, кількість унікальних джерел, кількість пакетів від одного джерела, частоту звернень до одного порту, кількість TCP SYN-пакетів без завершення з’єднання, кількість UDP-пакетів і кількість ICMP-запитів. Окремо може визначатися частка однотипного трафіку в загальному потоці. Якщо, наприклад, більшість пакетів у вікні становлять SYN-пакети або ICMP-запити, це може свідчити про підозрілий характер активності. Якщо ж трафік рівномірний і відповідає звичайному профілю роботи пристрою, він не розглядається як небезпечний.

У запропонованому алгоритмі важливим є те, що рішення приймається не за одним пакетом, а за поведінкою трафіку протягом певного часу. Це робить систему більш стійкою до випадкових коливань і дозволяє точніше оцінити

реальний стан мережі. Наприклад, один SYN-пакет або один ICMP-запит не становить загрози. Небезпечною є ситуація, коли такі пакети надходять масово, повторюються, спрямовуються на один вузол або створюють помітне перевищення нормального рівня. Саме така логіка відповідає природі DDoS-атак, у яких шкідливість проявляється не в одному окремому пакеті, а в загальній інтенсивності потоку.

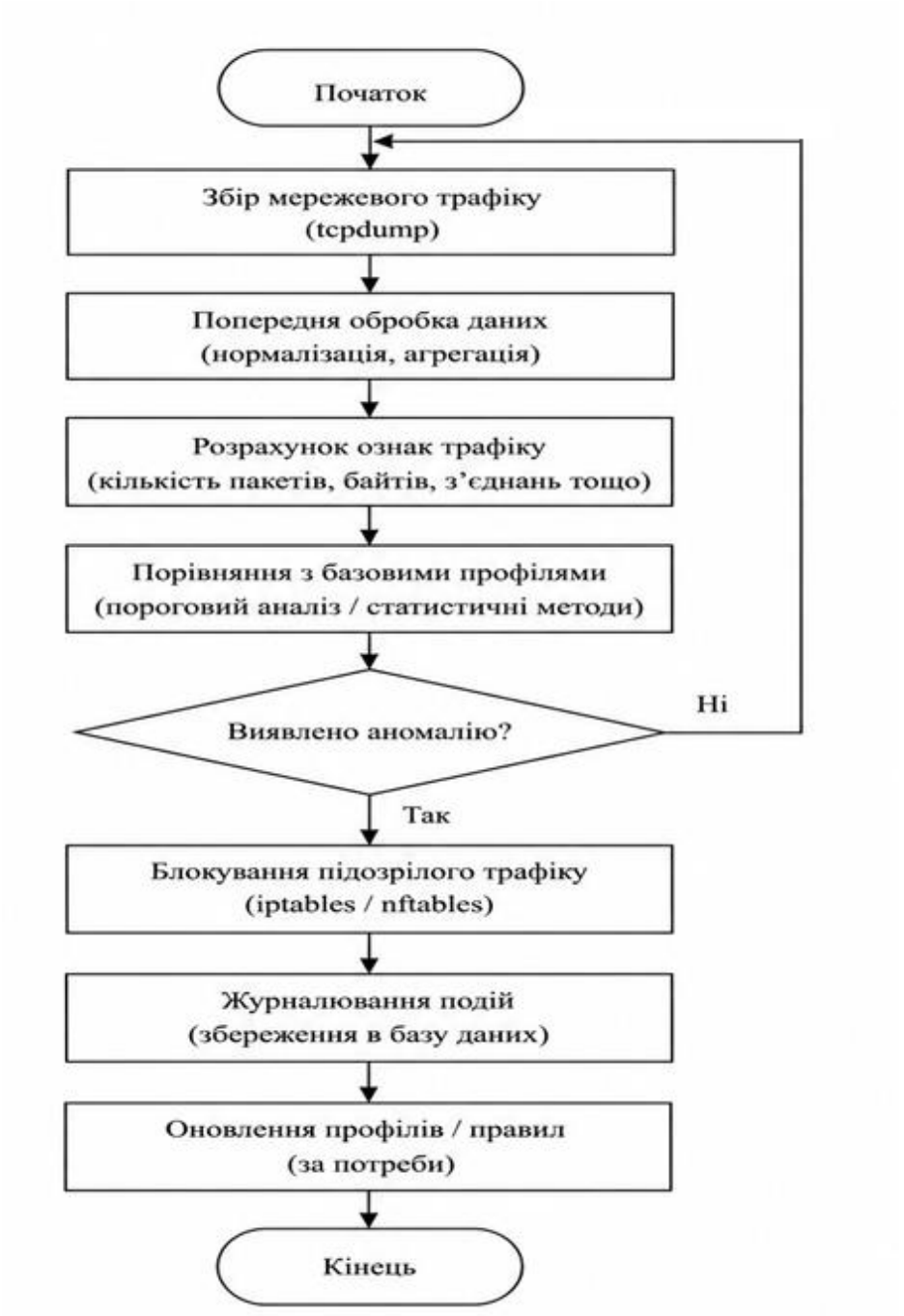


Рисунок 2.4 – Алгоритм виявлення аномального мережевого трафіку

Алгоритм також враховує можливість налаштування під конкретну мережу. Для цього порогові значення мають зберігатися в окремому конфігураційному файлі або базі даних. Такий підхід дозволяє змінювати допустимі межі без переписування програмного коду. Наприклад, для тестового сенсора можна встановити низький поріг кількості пакетів, а для локального вебсервера - вищий. Якщо під час перевірки виявиться, що система реагує занадто часто або, навпаки, пропускає підозрілу активність, пороги можна скоригувати. Це робить засіб більш гнучким і придатним до різних сценаріїв використання.

У межах практичного прототипу алгоритм може працювати у фоновому режимі. Після запуску програма постійно отримує параметри пакетів, оновлює статистику, перевіряє часові вікна та формує події. Адміністратор не повинен вручну запускати перевірку після кожного пакета, оскільки система працює безперервно. Водночас через вебінтерфейс або журнал можна переглядати, які події зафіксовано, які адреси визнано підозрілими та які правила застосовано. Така організація відповідає ідеї локального фільтра, який постійно контролює стан IoT-сегмента.

Під час розроблення алгоритму враховано обмеження Raspberry Pi. Оскільки ця платформа не є високопродуктивним сервером або спеціалізованим мережевим пристроєм, алгоритм не повинен виконувати надто складні обчислення для кожного пакета. Через це обрано легку модель аналізу, яка працює з лічильниками, часовими інтервалами та простими умовами перевищення порогів. Такий підхід забезпечує достатню швидкодію для локального стенда та не перевантажує систему зайвою обробкою. При цьому логіка залишається зрозумілою, що важливо для пояснення практичної частини бакалаврської роботи.

2.5 Розроблення механізму фільтрації та реагування на DDoS-активність

Механізм фільтрації та реагування на DDoS-активність є логічним продовженням алгоритму виявлення аномального мережевого трафіку. Якщо алгоритм аналізу визначає, що певний потік пакетів має ознаки підозрілої або небезпечної активності, то механізм фільтрації відповідає за практичну дію, яка має зменшити навантаження на захищений IoT-сегмент. У межах бакалаврської роботи цей механізм розроблено як частину апаратно-програмного засобу на базі Raspberry Pi, що працює між основною мережею та IoT-пристроями. Його основне призначення полягає у тому, щоб не лише фіксувати факт аномальної активності, а й своєчасно обмежувати трафік, який може призвести до втрати доступності пристроїв.

Необхідність такого механізму пояснюється тим, що саме по собі виявлення аномального трафіку ще не забезпечує захист. Якщо система лише записує подію до журналу, але не впливає на проходження пакетів, IoT-пристрій продовжує отримувати надмірне навантаження. Для малоресурсних пристроїв це є критичним, оскільки навіть короткий інтервал інтенсивного трафіку може призвести до затримок, зависання, розриву з'єднання або перезавантаження. Через це в структурі засобу передбачено окремий блок реагування, який отримує інформацію від модуля виявлення аномалій і перетворює її на конкретні правила фільтрації.

Основою механізму реагування є поетапна обробка події безпеки. Після того як алгоритм аналізу визначає перевищення допустимих порогів, формується запис про підозрілу активність. У цьому записі зазначається джерело трафіку, цільовий IoT-пристрій, тип протоколу, порт призначення, кількість пакетів, тривалість перевищення та рівень небезпеки. Саме ці дані використовуються для вибору подальшої дії. Якщо подія має низький рівень небезпеки, система може лише зафіксувати її у журналі. Якщо активність повторюється або має ознаки очевидного DDoS-навантаження, система переходить до активної фільтрації.

У запропонованому механізмі передбачено кілька варіантів реагування. Найпростішим є тимчасове блокування IP-адреси, з якої надходить надмірна кількість пакетів. Такий варіант добре підходить для ситуацій, коли аномальна активність чітко пов’язана з одним джерелом. Наприклад, якщо одна IP-адреса за короткий проміжок часу багаторазово надсилає TCP SYN-пакети до IoT-пристрою, фільтр може створити правило, яке відкидає пакети від цього джерела.

Іншим варіантом реагування є обмеження частоти пакетів. Такий підхід доцільний у випадках, коли повне блокування джерела не є бажаним або коли трафік може містити як легітимні, так і підозрілі запити. Наприклад, ICMP-запити можуть використовуватися для діагностики мережі, тому їх повна заборона не завжди є правильною.

У структурі механізму виділено кілька основних етапів. Спочатку модуль виявлення аномалій передає до блоку реагування сформовану подію. Далі система перевіряє рівень небезпеки та визначає тип трафіку. Після цього обирається дія: запис у журнал, обмеження частоти, тимчасове блокування IP-адреси або фільтрація певного протоколу чи порту. Потім формується правило для системного фільтра Linux, яке застосовується до мережевого потоку. Після виконання дії інформація зберігається в журналі та відображається у вебінтерфейсі. Така послідовність забезпечує зрозумілий і контрольований цикл реагування. Окремо передбачено ведення списку активних блокувань. Такий список потрібний для того, щоб система знала, які адреси або потоки вже обмежено, коли саме правило створено та коли воно має бути видалене. Без такого обліку можливе накопичення застарілих правил, які надалі можуть заважати нормальній роботі мережі. Тому модуль реагування має не лише додавати нові правила, а й періодично перевіряти їхній стан. Якщо термін дії тимчасового блокування завершився, правило видаляється, а відповідна подія записується до журналу. Це дозволяє підтримувати фільтр у актуальному стані.

На рисунку 2.5 показано механізм фільтрації та реагування на DDoS-активність, у якому відображено шлях від виявлення аномального трафіку до

створення правила блокування, запису події та відображення результату у вебінтерфейсі.

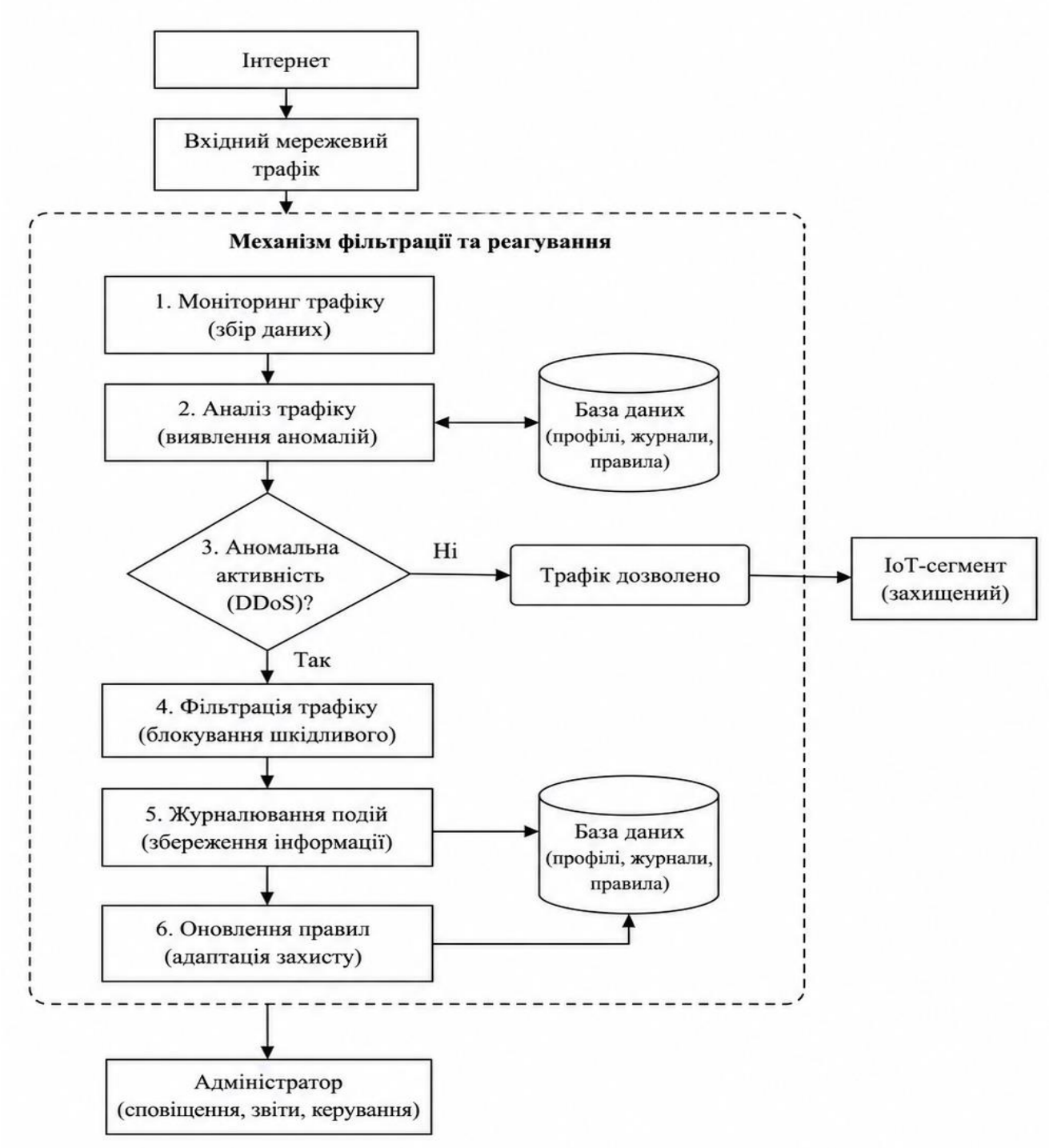


Рисунок 2.5 – Механізм фільтрації та реагування на DDoS-активність

2.6 Висновки до другого розділу

У другому розділі бакалаврської роботи розроблено структуру апаратно-програмного засобу захисту IoT-пристроїв від DDoS-атак. На основі результатів першого розділу сформовано загальну архітектуру системи, у якій центральним елементом виступає Raspberry Pi. Цей вузол розміщено між основною мережею та захищеним IoT-сегментом, що дозволяє контролювати трафік до його потрапляння на малоресурсні пристрої.

У розділі обґрунтовано вибір апаратних і програмних компонентів. Як апаратну основу використано Raspberry Pi, оскільки ця платформа підтримує Linux-середовище, має достатню продуктивність для локального аналізу трафіку, може працювати з мережевими інтерфейсами та дозволяє запускати власні програмні модулі. До складу стенда також включено маршрутизатор або комутатор, IoT-пристрій чи його емулятор, комп'ютер адміністратора та вузол для створення тестового навантаження.

Окремо розроблено структуру мережевого фільтра на базі Raspberry Pi. У ній виділено модуль зчитування трафіку, модуль попередньої обробки, блок аналізу, механізм виявлення аномалій, модуль реагування, журнал подій і засоби відображення результатів. Такий поділ дозволяє подати систему як послідовний набір взаємопов'язаних компонентів, де кожен модуль виконує окрему функцію.

У другому розділі також розроблено алгоритм виявлення аномального мережевого трафіку. Його основу становить аналіз активності за часовими вікнами та порівняння поточних показників із визначеними пороговими значеннями. Алгоритм враховує загальну кількість пакетів, кількість пакетів від одного джерела, активність за протоколами TCP, UDP та ICMP, кількість TCP SYN-пакетів, частоту звернень до портів і тривалість перевищення нормального рівня. Такий підхід є достатньо простим для реалізації на Raspberry Pi, але водночас придатним для виявлення базових проявів DDoS-активності в локальному IoT-сегменті.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ПЕРЕВІРКА РОБОТИ ЗАСОБУ ЗАХИСТУ

3.1 Формування експериментального стенда на базі Raspberry Pi

Практичну реалізацію апаратно-програмного засобу захисту IoT-пристроїв від DDoS-атак розпочато з формування експериментального стенда на базі Raspberry Pi. Саме стенд є основою для подальшої перевірки роботи програмних модулів, алгоритму виявлення аномального трафіку, механізму фільтрації та вебінтерфейсу моніторингу. У межах бакалаврської роботи експериментальний стенд сформовано як локальне мережеве середовище, у якому Raspberry Pi виконує роль проміжного фільтрувального вузла між основною мережею та захищеним IoT-сегментом. Такий підхід дозволяє не лише описати архітектуру засобу теоретично, а й показати її роботу в умовах, наближених до реального використання.

Основна ідея побудови стенда полягає в тому, що IoT-пристрій або його програмний емулятор не підключається до основної мережі напряму. Перед ним розміщено Raspberry Pi, який приймає або контролює трафік, спрямований до захищеного вузла. У цій схемі Raspberry Pi виконує функції локального шлюзу, аналізатора трафіку, засобу фільтрації та вузла журналювання подій. Саме через нього проходить мережевий потік, який далі може бути пропущений до IoT-сегмента або обмежений у разі виявлення ознак DDoS-активності. Така побудова стенда дає змогу перевірити повний цикл роботи системи: надходження пакетів, аналіз параметрів, виявлення аномалії, створення правила фільтрації та відображення результату.

До складу експериментального стенда включено Raspberry Pi, маршрутизатор або комутатор, комп’ютер адміністратора, захищений IoT-пристрій або його емулятор, а також окремий вузол для створення тестового навантаження. Raspberry Pi виступає центральним компонентом стенда, оскільки саме на ньому розгорнуто програмну частину засобу захисту. Маршрутизатор

або комутатор забезпечує фізичне з'єднання між елементами мережі. Комп'ютер адміністратора використано для віддаленого доступу до Raspberry Pi, налаштування програмних модулів, перегляду журналів і роботи з вебінтерфейсом. Захищений IoT-вузол імітує пристрій, доступність якого потрібно зберегти під час підвищеного або аномального трафіку. Окремий тестовий вузол потрібний для моделювання нормальних запитів і навантаження, схожого на DDoS-активність.

У практичній реалізації Raspberry Pi налаштовано як пристрій, що працює в Linux-середовищі та має доступ до мережевих інструментів. На плату встановлено операційну систему Raspberry Pi OS, після чого виконано базове налаштування мережевих параметрів, доступу через SSH, оновлення системних пакетів і підготовку середовища для запуску програмного забезпечення. Такий етап є необхідним, оскільки стабільність роботи фільтра залежить не лише від програмного коду, а й від правильного системного налаштування плати. Якщо мережеві інтерфейси, маршрутизація або права доступу налаштовані некоректно, програмна частина не зможе повноцінно зчитувати трафік і застосовувати правила фільтрації.

У стенді передбачено розділення мережі на дві умовні частини. Перша частина відповідає основній мережі, з якої можуть надходити запити до IoT-пристрою. Друга частина відповідає захищеному IoT-сегменту. Між ними розміщено Raspberry Pi. Така побудова дозволяє чітко визначити напрям руху трафіку та місце, у якому має виконуватися фільтрація. Якщо використовується два мережеві інтерфейси, один із них підключено до маршрутизатора або основної мережі, а другий - до IoT-сегмента. У разі застосування одного інтерфейсу логічне розмежування може виконуватися через підмережі та правила маршрутизації, однак у межах стенда зручнішим є варіант із двома інтерфейсами, оскільки він наочніше демонструє роботу фільтра як проміжного вузла.

У ролі захищеного IoT-пристрою може використовуватися реальний мікроконтролерний вузол, невеликий вебсервер, MQTT-клієнт або програмний емулятор. У межах бакалаврської роботи доцільно застосувати саме емулятор або простий локальний сервіс, оскільки він дозволяє контрольовано перевіряти реакцію системи без ризику пошкодження реального обладнання. Наприклад, на окремому пристрої або в локальному середовищі може бути запущено невеликий вебсервер, який відповідає на HTTP-запити та імітує роботу IoT-панелі. Також може бути використано простий MQTT-клієнт, який періодично передає короткі повідомлення. Такий підхід достатній для перевірки доступності пристрою до, під час і після активації фільтрації.

Окремий вузол для генерації тестового навантаження потрібний для перевірки поведінки стенда в різних режимах. У нормальному режимі цей вузол надсилає звичайні запити до IoT-пристрою з невеликою частотою. У режимі підвищеного навантаження кількість запитів поступово збільшується. У режимі моделювання DDoS-активності створюється інтенсивний потік однотипних пакетів, наприклад TCP SYN-запитів, UDP-пакетів або ICMP-запитів. У роботі не ставиться завдання створити реальну шкідливу атаку, тому тестове навантаження використовується лише в межах контрольованого лабораторного стенда. Його призначення полягає в тому, щоб перевірити, чи здатен фільтр виявити різке відхилення від нормальної поведінки та застосувати відповідне обмеження.

Після фізичного підключення компонентів виконано базову перевірку мережевої зв’язності. На цьому етапі перевірено, чи доступний Raspberry Pi з комп’ютера адміністратора, чи правильно визначаються мережеві інтерфейси, чи проходить трафік між основною мережею та IoT-сегментом, а також чи доступний захищений пристрій через проміжний вузол. Така перевірка потрібна для того, щоб відокремити проблеми фізичного підключення від помилок програмної логіки. Якщо пристрої не бачать один одного на рівні мережі,

подальша перевірка алгоритму виявлення аномалій не матиме практичного сенсу.

На рисунку 3.1 показано структуру експериментального стенда, у якому Raspberry Pi розміщено між основною мережею, вузлом генерації навантаження та захищеним IoT-сегментом.

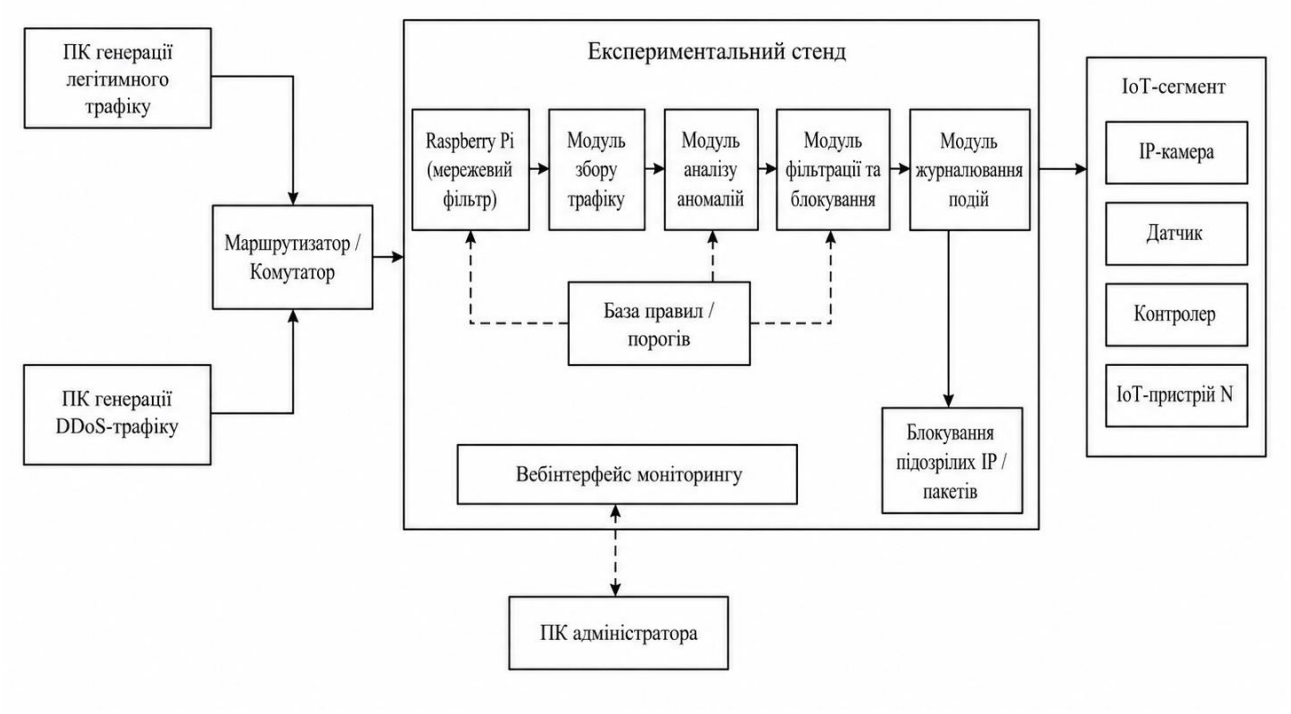


Рисунок 3.1 – Структура експериментального стенда для перевірки захисту IoT-пристроїв

На Raspberry Pi налаштовано IP-адресацію відповідно до структури стенда. Для інтерфейсу, підключеного до основної мережі, задано адресу з відповідної підмережі маршрутизатора. Для інтерфейсу, підключеного до IoT-сегмента, задано окрему адресу, яка використовується як шлюз для захищених пристроїв. Така схема дозволяє направити трафік до IoT-вузла через Raspberry Pi та забезпечити можливість його фільтрації. Додатково налаштовано пересилання пакетів на рівні операційної системи, оскільки Raspberry Pi має не лише приймати трафік, а й передавати його між мережевими сегментами.

У межах стенда також передбачено кілька сценаріїв перевірки. Перший сценарій відповідає нормальному режиму, коли IoT-пристрій отримує невелику кількість запитів і стабільно відповідає на них. Другий сценарій передбачає поступове збільшення трафіку, щоб перевірити, як система реагує на наближення до порогових значень. Третій сценарій імітує аномальну активність, коли кількість пакетів різко зростає або один тип трафіку починає переважати над іншими. У кожному сценарії фіксується стан захищеного пристрою, кількість оброблених пакетів, реакція Raspberry Pi та записи в журналі подій.

Важливою частиною формування стенда є визначення нормального профілю трафіку. Перед запуском тестового навантаження система працює певний час у режимі спостереження. У цей період фіксується, скільки пакетів зазвичай надходить до IoT-пристрою, які протоколи використовуються, які порти є активними та з яких джерел надходять легітимні запити. На основі такого спостереження можна точніше налаштувати порогові значення. Якщо пороги встановити без урахування реальної поведінки стенда, система може або надто часто спрацьовувати на нормальну активність, або запізно реагувати на підозрілий трафік.

Під час формування стенда враховано також безпечність виконання тестів. Усі перевірки проводяться в локальній контрольованій мережі, без спрямування навантаження на сторонні ресурси. Тестовий трафік використовується лише для перевірки власного стенда та захищеного вузла. Це дозволяє уникнути небажаного впливу на зовнішні системи й водночас отримати достатньо даних для оцінювання роботи фільтра. У такому форматі моделювання DDoS-активності має навчальний і демонстраційний характер, а не створює реальної загрози для інших мереж.

Сформований експериментальний стенд дозволяє послідовно перевірити всі ключові компоненти апаратно-програмного засобу. На рівні апаратної частини перевіряється стабільність Raspberry Pi, робота мережевих інтерфейсів і зв’язок між сегментами. На рівні системного програмного забезпечення

перевіряється маршрутизація, пересилання пакетів і робота засобів фільтрації. На рівні прикладної програми перевіряється зчитування трафіку, аналіз показників, виявлення аномалій, запис подій і взаємодія з вебінтерфейсом. Така поетапність дозволяє чітко визначити, який саме компонент відповідає за кожну частину роботи системи.

Практична цінність сформованого стенда полягає в тому, що він відображає реалістичну ситуацію невеликої IoT-мережі. У багатьох побутових, навчальних або малих офісних середовищах IoT-пристрої підключаються до локальної мережі та не мають складного захисту. Розміщення окремого фільтрувального вузла перед ними дозволяє підсилити контроль трафіку без заміни самих пристроїв. У межах бакалаврської роботи це дає змогу показати не абстрактну модель, а практичний варіант побудови захисту, який можна реалізувати доступними засобами.

3.2 Реалізація програмного модуля збору та аналізу мережевого трафіку

Після формування експериментального стенда на базі Raspberry Pi реалізовано програмний модуль збору та аналізу мережевого трафіку. Цей модуль є однією з основних частин апаратно-програмного засобу захисту IoT-пристроїв від DDoS-атак, оскільки саме він забезпечує отримання первинної інформації про пакети, що надходять до захищеного IoT-сегмента. Без такого модуля система не може визначити, які джерела створюють навантаження, які протоколи використовуються, до яких портів надходять звернення та чи перевищує поточна активність нормальний рівень для локальної мережі.

Програмний модуль реалізовано на Raspberry Pi в середовищі операційної системи Linux. Такий вибір пов’язаний із тим, що Raspberry Pi підтримує стандартні мережеві інструменти, дозволяє працювати з інтерфейсами Ethernet, запускати фонові служби, використовувати Python-скрипти та взаємодіяти із системними засобами фільтрації. У межах практичної реалізації модуль не

виконує повне збереження вмісту всіх пакетів, оскільки для виявлення DDoS-активності головне значення мають не дані користувача, а службові параметри трафіку. Це зменшує навантаження на Raspberry Pi та робить роботу засобу стабільнішою.

Основним завданням модуля є зчитування параметрів пакетів, які проходять через мережевий фільтр або надходять на інтерфейс Raspberry Pi. До таких параметрів належать IP-адреса джерела, IP-адреса призначення, тип протоколу, порт джерела, порт призначення, час надходження пакета, розмір пакета та напрям руху трафіку. Для TCP-пакетів додатково враховано службові прапорці, зокрема SYN, оскільки саме вони мають важливе значення для виявлення SYN flood-активності. Для UDP-пакетів основну увагу приділено кількості пакетів, порту призначення та повторюваності звернень. Для ICMP-трафіку зафіксовано частоту діагностичних запитів до захищеного вузла.

На рисунку 3.2 показано структуру програмного модуля збору та аналізу мережевого трафіку, у якій відображено послідовність переходу від отримання пакетів до формування статистичних показників і передачі результатів до модуля виявлення аномалій.

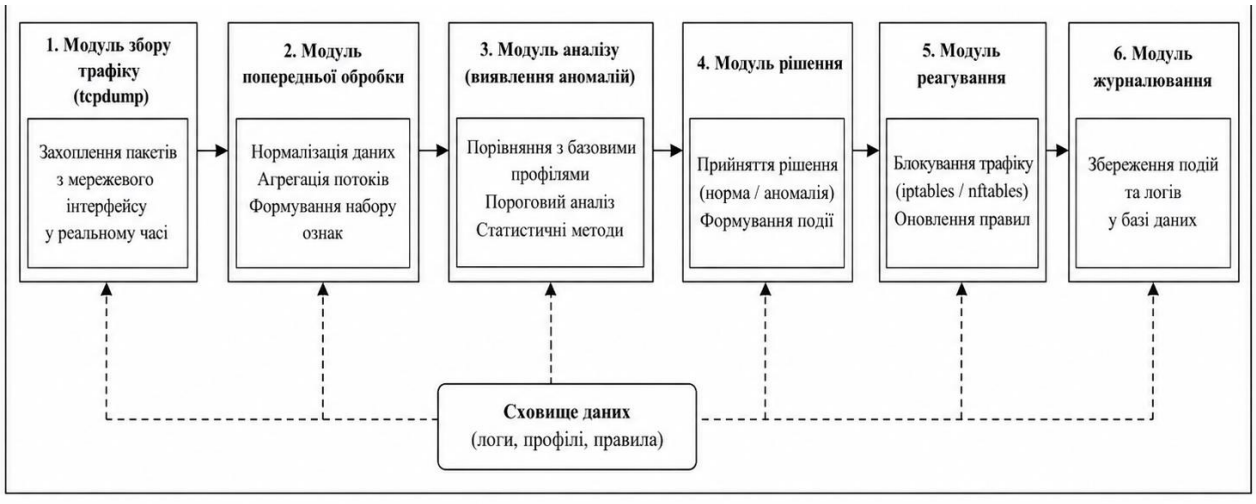


Рисунок 3.2 – Структура програмного модуля збору та аналізу трафіку

У структурі модуля першим етапом є підключення до мережевого інтерфейсу Raspberry Pi. Саме через цей інтерфейс проходить трафік між основною мережею та IoT-сегментом. На цьому рівні програмна частина отримує доступ до потоку пакетів і починає фіксувати їхні службові характеристики. Для практичної реалізації використано підхід, за якого модуль працює у фоновому режимі та постійно спостерігає за трафіком. Це дозволяє не запускати перевірку вручну після кожної зміни навантаження, а забезпечити безперервний контроль мережевої активності.

Після отримання пакета виконується первинне виділення службових полів. На цьому етапі з пакета відбираються тільки ті дані, які потрібні для подальшого аналізу. Такий підхід є важливим, оскільки повне збереження або глибоке розбирання кожного пакета створило б зайве навантаження на Raspberry Pi. У межах реалізованого модуля достатньо зафіксувати, звідки надійшов пакет, куди він спрямований, який протокол використано та до якого порту відбулося звернення. Саме ці параметри надалі стають основою для визначення підозрілої поведінки.

Наступним етапом є групування пакетів за часовими інтервалами. У модулі реалізовано обробку трафіку не як нескінченного потоку окремих пакетів, а як набору показників за короткі проміжки часу. Такий принцип дозволяє оцінити не один випадковий пакет, а поведінку трафіку в межах певного часового вікна. Наприклад, система може підрахувати, скільки пакетів надійшло за одну секунду, скільки з них належали до TCP, UDP або ICMP, скільки звернень виконано до одного порту та яка IP-адреса проявила найбільшу активність. Завдяки цьому аналіз стає більш змістовним і придатним для виявлення аномальних навантажень.

Окремо реалізовано підрахунок активності за IP-адресами джерел. Для кожної адреси модуль формує коротку статистику: загальну кількість пакетів, кількість звернень до конкретного IoT-пристрою, кількість звернень до окремого порту та частоту повторення запитів. Така інформація потрібна для визначення

джерел, які створюють надмірне навантаження. Якщо одна IP-адреса за короткий проміжок часу надсилає значно більше пакетів, ніж інші, система може позначити її як потенційно підозрілу. У подальшому ці дані передаються до алгоритму виявлення аномальної активності.

Крім аналізу окремих джерел, у модулі передбачено оцінювання навантаження на цільовий IoT-пристрій. Це важливо, оскільки DDoS-активність не завжди надходить з однієї адреси. У розподіленому сценарії кожне джерело може надсилати порівняно невелику кількість пакетів, але сумарно вони створюють значне навантаження на захищений вузол. Тому модуль окремо підраховує загальну кількість пакетів, спрямованих до кожного IoT-пристрою. Якщо навантаження на конкретний вузол різко зростає, система отримує підставу для подальшої перевірки навіть тоді, коли окреме джерело не виглядає критичним.

Для TCP-трафіку реалізовано окрему перевірку SYN-пакетів. Під час нормального встановлення TCP-з’єднання SYN-пакет є лише першим етапом взаємодії. Якщо таких пакетів надходить надто багато, а подальший обмін не має нормального характеру, це може свідчити про спробу перевантаження пристрою або сервісу. У модулі передбачено підрахунок SYN-пакетів за часовими вікнами та за IP-адресами джерел. Такі показники надалі використовуються для виявлення SYN flood-активності та формування рішення щодо блокування або обмеження підозрілого джерела.

Для UDP-трафіку реалізовано підрахунок кількості пакетів до конкретних портів. Оскільки UDP не вимагає встановлення з’єднання, небажаний потік таких пакетів може швидко створити навантаження на пристрій або мережевий канал. Програмний модуль фіксує, які порти отримують найбільшу кількість UDP-пакетів і чи відповідає така активність нормальному режиму роботи IoT-сегмента. Якщо UDP-пакети надходять до порту, який не використовується в межах стенда, або їхня кількість різко збільшується, ця інформація передається до наступного етапу аналізу як потенційна ознака аномального трафіку.

Для ICMP-трафіку реалізовано контроль частоти запитів. У звичайному режимі ICMP використовується переважно для перевірки доступності вузла, тому його кількість у невеликій IoT-мережі не має бути значною. Якщо Raspberry Pi фіксує велику кількість ICMP echo request-пакетів до захищеного пристрою, така активність може бути ознакою ICMP flood або попередньої перевірки доступності вузла. У модулі передбачено підрахунок таких запитів за короткими інтервалами, що дозволяє швидко визначити неприродне зростання діагностичного трафіку.

Для збереження проміжних результатів у програмному модулі використано внутрішні структури даних, у яких накопичується статистика за поточне часове вікно. Після завершення інтервалу сформовані показники передаються до блоку аналізу, а лічильники оновлюються для наступного циклу. Така організація дозволяє уникнути необмеженого накопичення даних у пам’яті Raspberry Pi. Система працює циклічно: збирає параметри пакетів, групує їх, підраховує показники, передає результат і очищує тимчасові значення. Це робить модуль придатним для тривалої роботи без суттєвого зростання використання пам’яті.

Важливою частиною реалізації є фільтрація зайвих або службових даних. Не весь трафік, який проходить через Raspberry Pi, має однакове значення для аналізу. Наприклад, службові пакети самої системи, звернення адміністратора або локальні запити можуть бути віднесені до довіреної активності. Для цього в програмному модулі передбачено можливість врахування дозволених адрес або виключень. Такі адреси не обов’язково повністю ігноруються, але їхня активність може оцінюватися окремо, щоб уникнути помилкового визначення адміністративного трафіку як атаки.

Результатом роботи модуля збору є не кінцеве рішення про блокування, а підготовлений набір статистичних показників. Це принципово важливо, оскільки модуль збору не повинен одночасно виконувати всі функції засобу захисту. Його завдання полягає в тому, щоб надати достовірні та впорядковані дані для

наступного блоку. Уже модуль виявлення аномалій порівнює ці показники з пороговими значеннями та визначає рівень небезпеки. Такий поділ спрощує програмну структуру й дозволяє окремо перевіряти правильність зчитування трафіку та правильність логіки прийняття рішень.

Після перевірки нормального режиму модуль протестовано під час підвищеної активності. У цьому режимі кількість запитів до захищеного пристрою поступово збільшується, а система фіксує зміну мережевих показників. Це дозволяє перевірити, чи правильно працюють лічильники, чи не втрачаються дані під час швидкого надходження пакетів і чи коректно оновлюються часові вікна. Саме на цьому етапі підтверджено, що модуль здатний відстежувати не лише одиничні запити, а й динаміку зміни трафіку.

3.3 Реалізація алгоритму виявлення аномальної активності

Після реалізації програмного модуля збору та попереднього аналізу мережевого трафіку виконано реалізацію алгоритму виявлення аномальної активності. Цей алгоритм є основним логічним ядром апаратно-програмного засобу захисту IoT-пристроїв від DDoS-атак, оскільки саме він перетворює набір зібраних мережевих показників на рішення про нормальний, підозрілий або небезпечний стан трафіку. Якщо модуль збору лише фіксує параметри пакетів, то алгоритм виявлення вже оцінює поведінку трафіку в часі, порівнює її з допустимими межами та визначає, чи потрібно передавати подію до механізму реагування.

Реалізований алгоритм побудовано з урахуванням специфіки IoT-сегмента. Для таких мереж характерний відносно передбачуваний трафік: датчики, контролери, шлюзи або прості вебінтерфейси зазвичай не створюють великої кількості одночасних з’єднань і не приймають значного потоку однотипних пакетів у нормальному режимі. Саме тому виявлення аномальної активності реалізовано на основі порівняння поточних показників із

встановленими порогами. Такий підхід не потребує складного машинного навчання, не перевантажує Raspberry Pi і водночас дозволяє фіксувати характерні прояви DDoS-навантаження.

У програмній логіці алгоритму використано оброблення трафіку за часовими вікнами. Це означає, що система не приймає рішення після кожного окремого пакета, а накопичує статистику за короткий проміжок часу. У межах одного такого інтервалу підраховано кількість пакетів, кількість унікальних IP-адрес джерел, кількість звернень до конкретного IoT-пристрою, активність за протоколами TCP, UDP та ICMP, кількість TCP SYN-пакетів і кількість звернень до окремих портів. Після завершення часового вікна сформовані значення передаються до блоку перевірки порогів, де визначається рівень підозрілості поточної активності.

У системі передбачено кілька груп ознак, за якими виконується виявлення аномального трафіку. Першою ознакою є різке збільшення загальної кількості пакетів за одиницю часу. Якщо трафік до IoT-пристрою раптово зростає у кілька разів порівняно з нормальним режимом, алгоритм фіксує таке відхилення як потенційно небезпечне. Другою ознакою є висока активність однієї IP-адреси. Якщо одне джерело протягом короткого проміжку часу надсилає багато пакетів до захищеного вузла, така поведінка може відповідати спробі перевантаження пристрою. Третьою ознакою є велика кількість однотипного трафіку, наприклад TCP SYN-пакетів, UDP-пакетів до одного порту або ICMP-запитів.

Окремо реалізовано перевірку TCP SYN-активності. У нормальному режимі SYN-пакети є початком встановлення TCP-з’єднання, тому їхня кількість не повинна різко переважати над іншими пакетами. Якщо система фіксує велику кількість SYN-пакетів від одного джерела або до одного IoT-пристрою, алгоритм підвищує рівень підозрілості цієї активності. Такий підхід дозволяє виявляти ознаки SYN flood-атаки, під час якої цільовий вузол отримує багато запитів на встановлення з’єднання, але не отримує нормального завершення сесій. Для

малоресурсного IoT-пристрою така ситуація є небезпечною, оскільки вона може швидко перевантажити мережевий стек.

Для UDP-трафіку реалізовано іншу логіку перевірки. Оскільки UDP не передбачає встановлення з’єднання, алгоритм не аналізує стан сесій, а оцінює кількість пакетів, повторюваність джерел і порти призначення. Якщо до одного порту надходить велика кількість UDP-пакетів, особливо якщо цей порт не використовується в нормальній роботі стенда, система позначає такий потік як підозрілий. Якщо UDP-пакети надходять із багатьох джерел, але спрямовані на один IoT-вузол, алгоритм враховує сумарне навантаження на цільовий пристрій. Це дозволяє виявляти не лише активність одного джерела, а й розподілений характер навантаження.

Для ICMP-трафіку реалізовано підрахунок кількості echo request-запитів за часовий інтервал. У звичайній локальній мережі ICMP використовується переважно для діагностики, тому його інтенсивність не має бути високою. Якщо кількість ICMP-запитів різко зростає, алгоритм фіксує це як ознаку можливої ICMP flood-активності або попередньої перевірки доступності вузла. При цьому ICMP-трафік не вважається шкідливим автоматично, оскільки він може використовуватися адміністратором для перевірки з’єднання. Через це в алгоритмі враховано не сам факт появи ICMP-пакетів, а саме їхню кількість, повторюваність і тривалість перевищення нормального рівня.

У реалізованому алгоритмі використано три основні стани трафіку. Перший стан відповідає нормальній роботі, коли всі показники залишаються в допустимих межах.

На рисунку 3.3 показано блок-схему реалізації алгоритму виявлення аномальної активності, у якій відображено послідовність оброблення статистики трафіку, перевірки порогових значень, визначення рівня небезпеки та передавання події до модуля реагування.

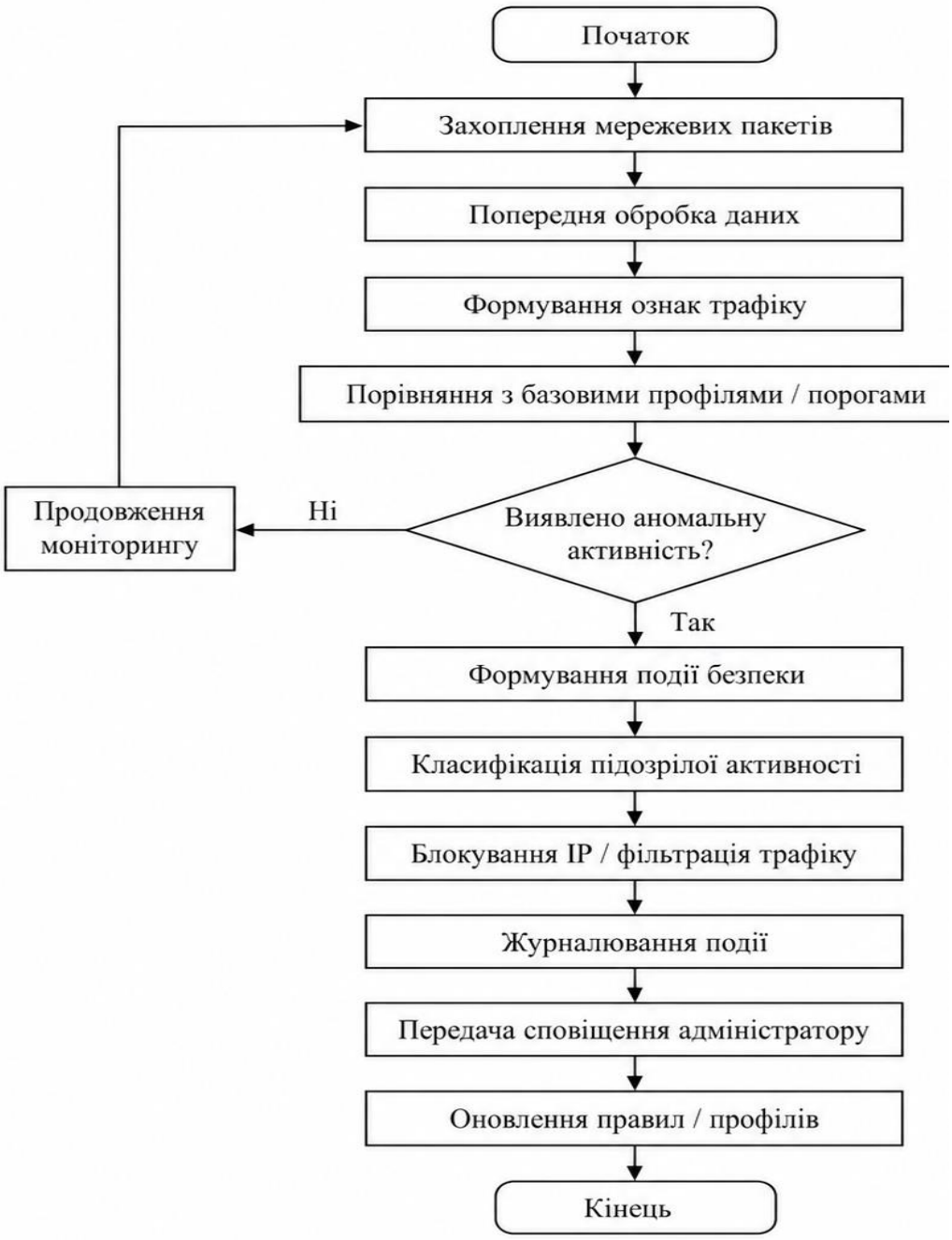


Рисунок 3.3 – Блок-схема реалізації алгоритму виявлення аномальної активності

У цьому випадку система лише оновлює статистику та не формує подію безпеки. Другий стан відповідає підозрілій активності, коли один або кілька показників наближаються до порогових значень або короткочасно їх перевищують. У такому режимі система записує попереджувальну інформацію до журналу, але не завжди передає команду на блокування. Третій стан

відповідає аномальній активності, коли перевищення повторюється кілька часових вікон поспіль або одразу має явно небезпечний характер. У цьому випадку формується подія для механізму фільтрації.

Для зменшення кількості помилкових спрацювань реалізовано перевірку тривалості відхилення. Якщо певний показник перевищив поріг лише один раз, але в наступному часовому вікні повернувся до нормального рівня, алгоритм не виконує активного реагування. Така ситуація може виникати під час перезапуску пристрою, звернення адміністратора до вебінтерфейсу або короткочасної службової активності. Якщо ж перевищення повторюється, рівень небезпеки поступово підвищується. Це дозволяє відрізняти випадкові піки від реального аномального навантаження.

У програмній частині для кожного джерела трафіку сформовано окремі лічильники. Вони зберігають кількість пакетів, кількість TCP SYN-запитів, UDP-пакетів, ICMP-запитів і звернень до окремих портів у межах поточного часового вікна. Після завершення інтервалу ці значення порівнюються з налаштованими межами. Якщо адреса перевищує допустимі показники, для неї формується ознака підозрілості. Якщо така адреса повторно створює перевищення, вона передається до модуля реагування як кандидат на тимчасове блокування або обмеження частоти трафіку.

Крім аналізу окремих IP-адрес, реалізовано оцінювання сумарного навантаження на IoT-пристрій. Це важливо для ситуацій, коли атака має розподілений характер. У такому випадку кожне окреме джерело може не перевищувати встановлений поріг, але загальна кількість пакетів до захищеного вузла різко зростає. Алгоритм фіксує таку ситуацію через лічильник трафіку до конкретної цільової адреси. Якщо сумарне навантаження на IoT-пристрій перевищує допустиму межу, система формує подію навіть тоді, коли окремі джерела виглядають не надто активними.

У реалізації передбачено використання конфігураційних параметрів. Порогові значення не закріплено жорстко в програмному коді, а винесено в

окремий файл або блок налаштувань. Це дозволяє змінювати межі спрацювання без переписування основної логіки. Наприклад, для простого сенсорного вузла встановлено нижчі пороги, оскільки він не повинен отримувати великий обсяг трафіку. Для вебінтерфейсу або тестового IoT-шлюзу межі можуть бути вищими, адже такі вузли здатні обробляти більше запитів. Така гнучкість робить алгоритм придатним для різних сценаріїв використання в межах одного стенда.

Перевірку реалізованого алгоритму виконано у кілька етапів. Спочатку система працювала в нормальному режимі, коли IoT-пристрій або його емулятор отримував невелику кількість звичайних запитів. На цьому етапі алгоритм не формував небезпечних подій, а лише оновлював статистику. Далі кількість запитів поступово збільшувалася, що дозволило перевірити роботу попереджувального рівня. Після цього змодельовано інтенсивніший потік однотипних пакетів, за якого система зафіксувала перевищення порогів і сформувала подію для модуля реагування. Така послідовність підтвердила, що алгоритм коректно відрізняє звичайну активність від аномальної.

Результати роботи алгоритму відображаються у журналі подій і вебінтерфейсі. У журналі зберігається інформація про час спрацювання, джерело трафіку, тип аномалії та рівень небезпеки. У вебінтерфейсі відображається поточний стан системи, кількість зафіксованих підозрілих подій, активні джерела навантаження та загальна інтенсивність трафіку. Це дозволяє візуально оцінити, як саме алгоритм реагує на зміну мережевої активності, і використати ці результати під час подальшого аналізу працездатності засобу.

3.4 Реалізація механізму автоматичного блокування підозрілого трафіку

Після реалізації алгоритму виявлення аномальної активності виконано практичну реалізацію механізму автоматичного блокування підозрілого трафіку. Цей механізм є важливою частиною апаратно-програмного засобу захисту IoT-пристроїв від DDoS-атак, оскільки саме він забезпечує перехід від пасивного

спостереження за мережею до активної протидії небажаному навантаженню. Якщо попередній модуль лише визначає факт перевищення порогових значень і формує подію безпеки, то модуль блокування виконує конкретну дію: обмежує або відкидає трафік, який може негативно вплинути на роботу захищеного IoT-сегмента.

Основна логіка автоматичного блокування побудована навколо подій, які надходять від алгоритму виявлення аномальної активності. Кожна така подія містить інформацію про джерело трафіку, цільовий пристрій, тип протоколу, перевищений показник, рівень небезпеки та рекомендований спосіб реагування. На основі цих даних модуль блокування визначає, яку дію потрібно застосувати. Якщо подія має попереджувальний характер, система може лише зафіксувати її в журналі. Якщо ж трафік має стійкі ознаки DDoS-активності, модуль створює правило фільтрації, яке тимчасово блокує або обмежує підозрілий потік.

Для практичної реалізації механізму блокування використано системні засоби фільтрації Linux, доступні на Raspberry Pi. У межах прототипу доцільно застосовувати iptables або nftables, оскільки ці інструменти дозволяють керувати проходженням пакетів на рівні операційної системи. За їх допомогою можна блокувати IP-адреси, обмежувати звернення до окремих портів, фільтрувати певні протоколи та відкидати пакети, які відповідають заданим умовам. У цій роботі механізм реалізовано так, щоб програмний модуль міг автоматично формувати команди для системного фільтра після отримання підтвердженої події безпеки.

У запропонованій системі блокування не застосовується після кожного окремого пакета. Такий підхід міг би створити надмірне навантаження на Raspberry Pi та призвести до хаотичного створення правил. Натомість рішення приймається після аналізу активності в межах часового вікна. Якщо протягом одного або кількох інтервалів фіксується перевищення допустимої кількості пакетів, підвищена кількість TCP SYN-запитів, інтенсивний UDP-потік або

надмірна ICMP-активність, система переводить подію у стан підтвердженої аномалії. Лише після цього запускається механізм автоматичного реагування.

На рисунку 3.4 показано послідовність автоматичного блокування підозрілого трафіку, де відображено перехід від отримання події про аномалію до створення правила фільтрації, запису результату до журналу та відображення інформації у вебінтерфейсі.

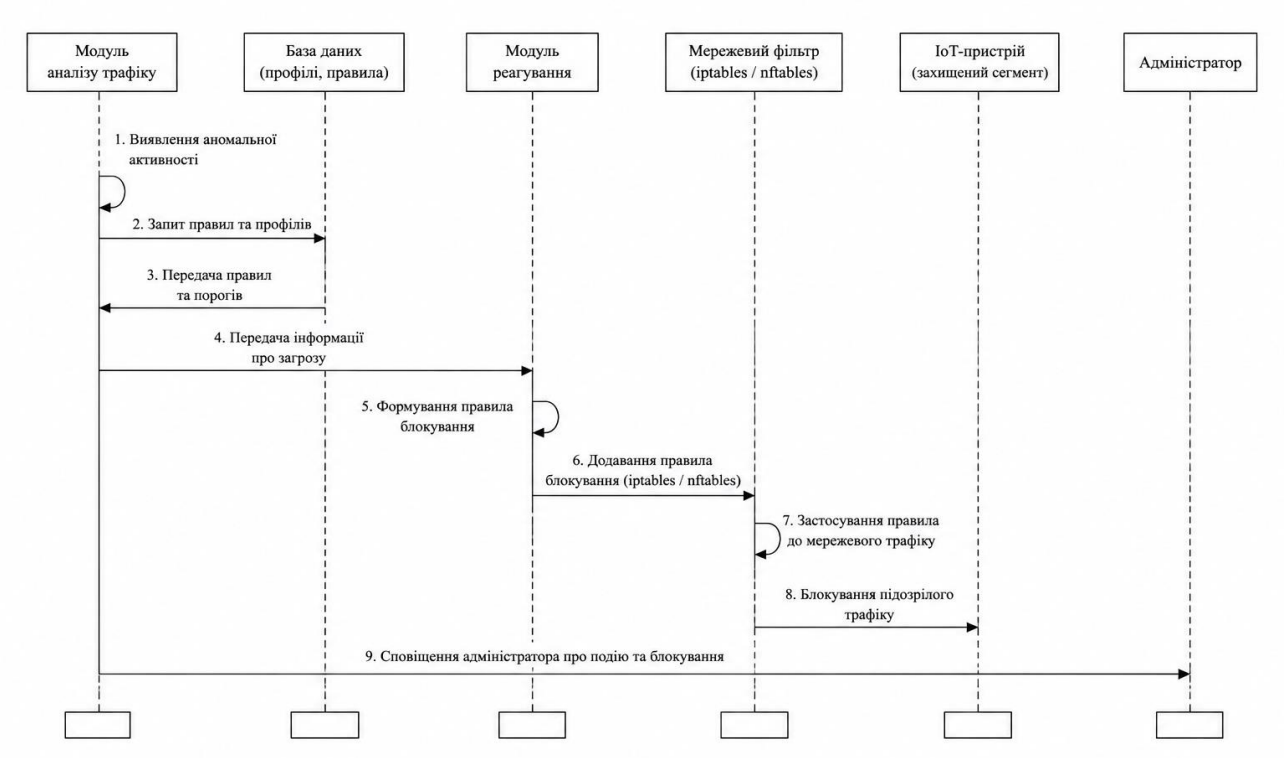


Рисунок 3.4 – Послідовність автоматичного блокування підозрілого трафіку

Першим етапом роботи механізму є отримання події від модуля виявлення аномальної активності. Подія надходить до блоку реагування разом із набором параметрів, які потрібні для прийняття рішення. Зокрема, система отримує IP-адресу джерела, IP-адресу захищеного IoT-пристрою, тип трафіку, кількість зафіксованих пакетів, перевищений поріг і рівень небезпеки. Якщо подія не містить достатніх даних для формування правила, вона лише записується до журналу як попередження. Якщо параметри дозволяють однозначно визначити

підозріле джерело або небезпечний тип трафіку, виконується підготовка правила блокування.

Другим етапом є перевірка списку винятків. У локальній мережі можуть бути службові вузли, які не повинні автоматично блокуватися без додаткової перевірки. До таких вузлів належить комп'ютер адміністратора, локальний сервер, шлюз або інший довірений пристрій. Якщо IP-адреса джерела входить до білого списку, система не створює правило повного блокування, а лише фіксує подію та позначає її як службову активність із підвищеним рівнем трафіку. Такий підхід дозволяє уникнути ситуації, коли адміністратор втрачає доступ до Raspberry Pi або вебінтерфейсу через помилкове спрацювання алгоритму.

Важливою частиною реалізації є журналювання кожної дії механізму блокування. До журналу записується не лише факт виявлення аномалії, а й конкретна реакція системи. Запис містить час події, адресу джерела, адресу цільового пристрою, тип трафіку, причину блокування, створене правило, тривалість дії та результат виконання команди. Якщо правило успішно застосовано, подія позначається як виконана. Якщо під час створення правила виникла помилка, вона також фіксується. Це дозволяє надалі перевірити, чи коректно система реагувала на підозрілий трафік і чи не виникали проблеми на рівні операційної системи.

Журнал подій використовується не тільки для технічної перевірки, а й для відображення результатів у вебінтерфейсі. У панелі моніторингу показано список активних блокувань, останні спрацювання, причини блокування, IP-адреси джерел і час завершення дії правил. Це робить роботу системи зрозумілою для адміністратора. Замість того щоб вручну переглядати системні таблиці фільтрації, достатньо відкрити вебінтерфейс і побачити, які адреси або типи трафіку обмежено. Такий підхід підвищує наочність практичної реалізації та полегшує демонстрацію роботи засобу.

Під час реалізації механізму блокування враховано потребу в ручному керуванні. Хоча основний режим роботи є автоматичним, адміністратор повинен

мати можливість переглянути активні правила, скасувати певне блокування, змінити порогові значення або тимчасово вимкнути автоматичне реагування. Це особливо важливо під час тестування, коли система налаштовується під конкретний стенд. Якщо виявлено, що фільтр занадто часто блокує легітимний трафік, пороги можуть бути скориговані, а автоматичне блокування тимчасово переведене в режим спостереження.

Перевагою реалізованого механізму є його простота та достатня гнучкість. Він не потребує складного спеціалізованого обладнання, оскільки використовує можливості Linux і програмну логіку на Raspberry Pi. Водночас він дозволяє реалізувати кілька типів реагування: блокування джерела, обмеження частоти пакетів, фільтрацію за портом або протоколом, запис події без активної дії та ручне скасування правил. Такий набір функцій є достатнім для локального IoT-сегмента й відповідає можливостям бакалаврського прототипу.

3.5 Реалізація вебінтерфейсу моніторингу стану захисту

Вебінтерфейс реалізовано як локальну панель керування, що відкривається з комп’ютера адміністратора через браузер. Такий підхід є зручним для експериментального стенда, оскільки не потребує встановлення додаткового клієнтського програмного забезпечення. Достатньо, щоб комп’ютер адміністратора мав доступ до Raspberry Pi у локальній мережі. У такому випадку адміністратор може переглядати стан фільтра, аналізувати останні події, контролювати активні блокування та оцінювати інтенсивність трафіку без безпосереднього підключення до консолі Raspberry Pi. Це робить взаємодію із засобом простішою та зрозумілішою.

Основною метою вебінтерфейсу є відображення інформації, яка формується під час роботи модулів збору, аналізу, виявлення аномальної активності та блокування підозрілого трафіку. Інтерфейс не виконує самостійного аналізу пакетів, а отримує вже підготовлені дані з програмної

частини системи. До таких даних належать поточна кількість оброблених пакетів, кількість активних підозрілих джерел, список заблокованих IP-адрес, типи зафіксованих аномалій, час останнього спрацювання, стан правил фільтрації та загальний режим роботи засобу. Завдяки цьому вебінтерфейс виступає не окремою системою, а візуальним представленням внутрішньої логіки апаратно-програмного засобу.

Для реалізації серверної частини вебінтерфейсу доцільно використано легкий вебфреймворк на Python, наприклад Flask або FastAPI. Такий вибір добре узгоджується із загальною програмною логікою засобу, оскільки основні модулі збору та аналізу трафіку також реалізуються у Python-середовищі. Це спрощує обмін даними між модулями, роботу з локальною базою даних і формування сторінок інтерфейсу. Вебсервер запускається безпосередньо на Raspberry Pi та працює у локальній мережі, а доступ до нього здійснюється через IP-адресу фільтрувального вузла та визначений порт.

У структурі вебінтерфейсу передбачено головну сторінку моніторингу, на якій відображається загальний стан системи. На цій сторінці показано, чи активний модуль збору трафіку, чи працює алгоритм виявлення аномалій, чи увімкнено автоматичне блокування та чи доступні системні засоби фільтрації. Така інформація потрібна для швидкої оцінки працездатності засобу. Якщо один із модулів не працює, це одразу відображається в інтерфейсі, що дозволяє швидко зрозуміти, на якому етапі виникла проблема. Для бакалаврської роботи така сторінка є особливо важливою, оскільки вона демонструє, що система має не лише внутрішню логіку, а й зрозуміле відображення стану.

Окремий блок інтерфейсу присвячено поточній статистиці мережевого трафіку. У ньому виводиться кількість пакетів за останній часовий інтервал, загальна кількість оброблених пакетів, кількість TCP-, UDP- та ICMP-пакетів, а також кількість звернень до захищеного IoT-пристрою. Такі показники дозволяють побачити, як змінюється активність мережі в реальному або наближеному до реального часу. Якщо в нормальному режимі кількість пакетів

залишається невеликою, а під час тестового навантаження різко зростає, це добре помітно саме через панель моніторингу.

На рисунку 3.5 показано вебінтерфейс моніторингу стану захисту, у якому відображено поточну статистику трафіку, журнал подій, список активних блокувань і загальний режим роботи апаратно-програмного засобу.

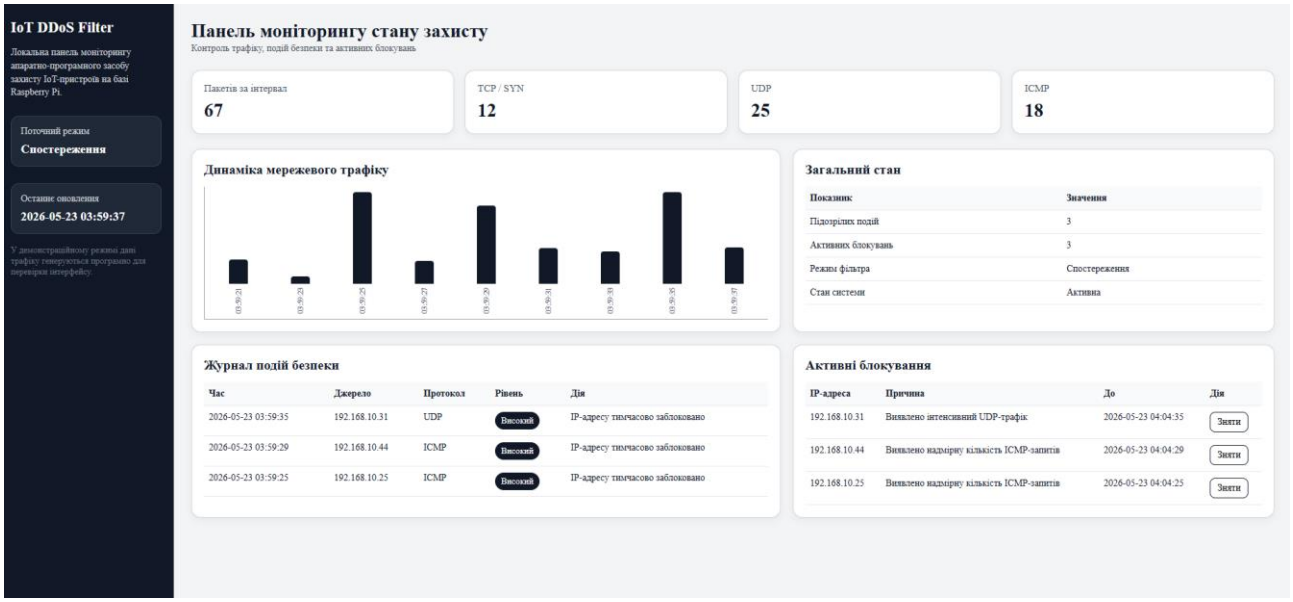


Рисунок 3.5 – Вебінтерфейс моніторингу мережевого трафіку та подій безпеки

Важливою частиною вебінтерфейсу є відображення журналу подій. У журналі подаються записи, які формуються під час роботи алгоритму виявлення аномальної активності та механізму автоматичного блокування. Кожен запис містить час події, IP-адресу джерела, адресу або назву цільового IoT-пристрою, тип трафіку, причину спрацювання, рівень небезпеки та застосовану дію. Завдяки цьому можна простежити, коли саме система зафіксувала підозрілу активність, чому вона визнала її небезпечною та яке рішення прийняла. Такий журнал має не лише інформаційне, а й аналітичне значення, оскільки на його основі в подальшому виконується оцінка результатів роботи засобу.

Список активних блокувань подано окремим блоком, оскільки він показує поточний стан механізму реагування. У ньому відображаються IP-адреси, для яких створено тимчасові правила фільтрації, причина блокування, час початку

дії правила та орієнтовний час його завершення. Така інформація потрібна для контролю за тим, які джерела на цей момент обмежені системою. Якщо підозріле джерело повторно створює навантаження, у списку може оновлюватися час блокування або змінюватися рівень небезпеки. Це дозволяє не лише фіксувати факт блокування, а й спостерігати за його динамікою.

У вебінтерфейсі також передбачено відображення режиму роботи системи. Засіб може перебувати в режимі спостереження, режимі попередження або режимі активного захисту. У режимі спостереження система лише збирає статистику й не створює правил блокування. У режимі попередження вона фіксує підозрілі події, але не завжди застосовує фільтрацію. У режимі активного захисту підтверджені аномалії призводять до автоматичного створення правил блокування або обмеження трафіку. Відображення режиму роботи є важливим, оскільки одна й та сама мережева активність може по-різному оброблятися залежно від поточного стану засобу.

Для зручності сприйняття інформацію в інтерфейсі згруповано за логічними блоками. Верхня частина сторінки може містити загальний статус системи та короткі числові показники. Центральна частина відображає поточну статистику трафіку та активні джерела навантаження.

Особливу увагу під час реалізації приділено тому, щоб вебінтерфейс не створював надмірного навантаження на Raspberry Pi. Оскільки плата одночасно виконує функції збору трафіку, аналізу, фільтрації та збереження подій, інтерфейс має бути легким. Через це не використано складні візуальні ефекти або важкі клієнтські компоненти. Основний акцент зроблено на таблицях, коротких інформаційних блоках і простому відображенні показників. Такий підхід забезпечує достатню швидкодію й не заважає основній функції засобу - захисту IoT-сегмента.

3.6 Висновки до третього розділу

У третьому розділі бакалаврської роботи виконано практичну реалізацію апаратно-програмного засобу захисту IoT-пристроїв від DDoS-атак. На першому етапі сформовано експериментальний стенд на базі Raspberry Pi, у якому фільтрувальний вузол розміщено між основною мережею та захищеним IoT-сегментом.

У практичній частині реалізовано програмний модуль збору та аналізу мережевого трафіку. Він забезпечує зчитування службових параметрів пакетів, зокрема IP-адрес джерела та призначення, протоколу, порту, часу надходження, розміру пакета та службових ознак TCP-з'єднання. Отримані дані групуються за часовими інтервалами, джерелами, протоколами, портами та цільовими пристроями.

Далі реалізовано алгоритм виявлення аномальної активності. Він аналізує поточні показники трафіку, порівнює їх із пороговими значеннями та визначає рівень небезпеки. У системі враховано нормальний режим, режим попередження та режим активної аномалії. Такий поділ дозволяє не блокувати трафік після кожного короткого відхилення, а реагувати на підтверджені або повторювані ознаки небезпечної активності.

У розділі також реалізовано механізм автоматичного блокування підозрілого трафіку. Після отримання події від алгоритму виявлення аномалій система перевіряє рівень небезпеки, враховує список довірених адрес і створює тимчасове правило фільтрації. Блокування може застосовуватися до конкретної IP-адреси, протоколу або порту, залежно від характеру виявленої активності. Передбачено журналювання кожної дії, збереження активних блокувань і автоматичне очищення застарілих правил. Це дозволяє підтримувати фільтр в актуальному стані та уникати накопичення непотрібних обмежень.

ВИСНОВКИ

У бакалаврській роботі вирішено задачу розроблення апаратно-програмного засобу для захисту IoT-пристроїв від DDoS-атак на базі Raspberry Pi. У процесі виконання роботи розглянуто особливості побудови IoT-мереж, визначено їхні основні слабкі місця та показано, що пристрої Інтернету речей часто мають обмежені апаратні ресурси, спрощену програмну логіку й недостатні вбудовані засоби захисту. Через це навіть порівняно невелике аномальне мережеве навантаження може негативно впливати на доступність IoT-вузлів, спричиняти затримки, втрату зв’язку або нестабільну роботу сервісів.

У першому розділі проаналізовано загальну характеристику DDoS-атак на IoT-інфраструктуру та визначено основні ознаки аномального мережевого трафіку. До таких ознак віднесено різке збільшення кількості пакетів за короткий проміжок часу, надмірну активність окремих IP-адрес, велику кількість однотипних TCP SYN-, UDP- або ICMP-пакетів, часті звернення до одного порту та тривале перевищення нормального рівня трафіку. Також розглянуто існуючі засоби захисту від DDoS-атак, зокрема хмарні сервіси, провайдерські рішення, міжмережеві екрани, IDS/IPS-системи, WAF-рішення, Linux-фільтри та локальні шлюзи захисту. На основі цього встановлено, що для невеликого IoT-сегмента найбільш доцільним є компактний локальний фільтр, який можна розгорнути без дорогого спеціалізованого обладнання.

У другому розділі розроблено загальну архітектуру апаратно-програмного засобу захисту IoT-пристроїв. Центральним вузлом системи обрано Raspberry Pi, який розміщується між основною мережею та захищеним IoT-сегментом. Обґрунтовано вибір апаратних і програмних компонентів, визначено структуру мережевого фільтра, сформовано алгоритм виявлення аномального трафіку та розроблено механізм фільтрації й реагування на DDoS-активність. Запропонована структура поєднує модуль збору параметрів трафіку, блок попередньої обробки, алгоритм аналізу, механізм створення правил блокування,

журнал подій і вебінтерфейс моніторингу. Така побудова дозволяє реалізувати повний цикл роботи засобу: отримання пакетів, аналіз їхніх характеристик, визначення підозрілої активності, застосування фільтрації та відображення результатів.

У третьому розділі виконано практичну реалізацію основних компонентів системи. Сформовано експериментальний стенд на базі Raspberry Pi, у якому фільтрувальний вузол розміщено між основною мережею та захищеним IoT-пристроєм або його емулятором. Реалізовано програмний модуль збору й аналізу мережевого трафіку, який фіксує IP-адреси, протоколи, порти, часові інтервали, кількість пакетів і службові ознаки TCP-з'єднань. На основі цих даних реалізовано алгоритм виявлення аномальної активності, який працює з часовими вікнами та пороговими значеннями.

Окремим результатом практичної частини стало створення вебінтерфейсу моніторингу стану захисту. Через нього відображається поточна статистика трафіку, журнал подій безпеки, список активних блокувань, стан алгоритму виявлення аномалій і режим роботи системи. Наявність вебінтерфейсу зробила засіб більш зручним для перевірки та демонстрації, оскільки результати роботи не обмежуються консольними повідомленнями або системними журналами. Це дозволяє наочно простежити, як система реагує на зміну мережевої активності та які дії виконує після виявлення підозрілого трафіку.

У результаті виконання бакалаврської роботи створено прототип апаратно-програмного засобу, який здатний аналізувати мережевий трафік у локальному IoT-сегменті, виявляти ознаки DDoS-активності, формувати події безпеки, застосовувати правила фільтрації та відображати стан системи у вебінтерфейсі. Розроблений засіб не є заміною провайдерського або хмарного DDoS-захисту, однак він є практичним рішенням для невеликої локальної мережі, навчального стенда або домашнього IoT-середовища. Його перевага полягає в доступності, простоті розгортання, використанні стандартних Linux-інструментів і можливості адаптації під конкретні умови роботи IoT-сегмента.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Tariq U., Ahmed I., Bashir A. K., Shaukat K. A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review. *Sensors*. 2023. Vol. 23, No. 8. 4117. DOI: 10.3390/s23084117

2. Vishwakarma R., Jain A. K. A survey of DDoS attacking techniques and defence mechanisms in the IoT network. *Telecommunication Systems*. 2020. Vol. 73, No. 1. P. 3–25. DOI: 10.1007/s11235-019-00599-z

3. Khan Z. A., Namin A. S. A Survey of DDOS Attack Detection Techniques for IoT Systems Using BlockChain Technology. *Electronics*. 2022. Vol. 11, No. 23. 3892. DOI: 10.3390/electronics11233892

4. Alahmadi A. A. та ін. DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions. *Electronics*. 2023. Vol. 12, No. 14. 3103. DOI: 10.3390/electronics12143103

5. Pakmehr A. та ін. DDoS attack detection techniques in IoT networks: a survey. *Cluster Computing*. 2024. DOI: 10.1007/s10586-024-04662-6

6. Gelgi M., Guan Y., Arunachala S., Rao M. S. S., Dragoni N. Systematic Literature Review of IoT Botnet DDOS Attacks and Evaluation of Detection Techniques. *Sensors*. 2024. Vol. 24, No. 11. 3571. DOI: 10.3390/s24113571

7. Bhayo J., Shah S. A., Hameed S., Ahmed A., Nasir J., Draheim D. Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks. *Engineering Applications of Artificial Intelligence*. 2023. Vol. 123. 106432. DOI: 10.1016/j.engappai.2023.106432

8. Wang J., Wang L. SDN-Defend: A Lightweight Online Attack Detection and Mitigation System for DDoS Attacks in SDN. *Sensors*. 2022. Vol. 22, No. 21. 8287. DOI: 10.3390/s22218287

9. Yaser A. L., Mousa H. M., Hussein M. Improved DDoS Detection Utilizing Deep Neural Networks and Feedforward Neural Networks as Autoencoder. *Future Internet*. 2022. Vol. 14, No. 8. 240. DOI: 10.3390/fi14080240

10. Mittal M., Kumar K., Behal S. Deep learning approaches for detecting DDoS attacks: a systematic review. *Soft Computing*. 2023. Vol. 27, No. 18. P. 13039–13075. DOI: 10.1007/s00500-021-06608-1

11. Neto E. C. P. та ін. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors*. 2023. Vol. 23, No. 13. 5941. DOI: 10.3390/s23135941

12. Alsaedi A., Moustafa N., Tari Z., Mahmood A., Anwar A. TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. *IEEE Access*. 2020. Vol. 8. P. 165130–165150. DOI: 10.1109/ACCESS.2020.3022862

13. Singh C., Jain A. K. A comprehensive survey on DDoS attacks detection & mitigation in SDN-IoT network. *e-Prime - Advances in Electrical Engineering, Electronics and Energy*. 2024. Vol. 8. 100543. DOI: 10.1016/j.prime.2024.100543

14. Snehi M., Bhandari A., Verma J. Foggier skies, clearer clouds: A real-time IoT-DDoS attack mitigation framework in fog-assisted software-defined cyber-physical systems. *Computers & Security*. 2024. Vol. 139. 103702. DOI: 10.1016/j.cose.2024.103702

15. Dash S. K. та ін. Enhancing DDoS attack detection in IoT using PCA. *Egyptian Informatics Journal*. 2024. Vol. 25. 100450. DOI: 10.1016/j.eij.2024.100450

16. Srinivasa Rao G. та ін. DDoSNet: Detection and prediction of DDoS attacks from realistic multidimensional dataset in IoT network environment. *Egyptian Informatics Journal*. 2024. Vol. 27, No. 3. 100526. DOI: 10.1016/j.eij.2024.100526

17. Aslam M. та ін. Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT. *Sensors*. 2022. Vol. 22, No. 7. 2697. DOI: 10.3390/s22072697

18. Khraisat A., Gondal I., Vamplew P., Kamruzzaman J. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*. 2019. Vol. 2. 20. DOI: 10.1186/s42400-019-0038-7

19. Stoyanova M., Nikoloudakis Y., Panagiotakis S., Pallis E., Markakis E. K. A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues. *IEEE Communications Surveys & Tutorials*. 2020. Vol. 22, No. 2. P. 1191–1221. DOI: 10.1109/COMST.2019.2962586

20. Omolara A. E. та ін. The internet of things security: A survey encompassing unexplored areas and new insights. *Computers & Security*. 2022. Vol. 112. 102494. DOI: 10.1016/j.cose.2021.102494

21. Gaur V., Kumar R. Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices. *Arabian Journal for Science and Engineering*. 2022. Vol. 47. P. 1353–1374. DOI: 10.1007/s13369-021-05947-3

22. Ismail M. I. та ін. A Machine Learning-Based Classification and Prediction Technique for DDoS Attacks. *IEEE Access*. 2022. Vol. 10. P. 21443–21454. DOI: 10.1109/ACCESS.2022.3152577

23. Gupta B. B., Chaudhary P., Chang X., Nedjah N. Smart defense against distributed Denial of service attack in IoT networks using supervised learning classifiers. *Computers & Electrical Engineering*. 2022. Vol. 98. 107726. DOI: 10.1016/j.compeleceng.2022.107726

24. Mihoub A., Ben Fredj O., Cheikhrouhou O., Derhab A., Krichen M. Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques. *Computers & Electrical Engineering*. 2022. Vol. 98. 107716. DOI: 10.1016/j.compeleceng.2022.107716

25. Almaraz-Rivera J. G., Perez-Diaz J. A., Cantoral-Ceballos J. A. Transport and Application Layer DDoS Attacks Detection to IoT Devices by Using Machine Learning and Deep Learning Models. *Sensors*. 2022. Vol. 22, No. 9. 3367. DOI: 10.3390/s22093367

26. Sharma D. K. та ін. Anomaly detection framework to prevent DDoS attack in fog empowered IoT networks. *Ad Hoc Networks*. 2021. Vol. 121. 102603. DOI: 10.1016/j.adhoc.2021.102603

27. Islam U. та ін. Detection of Distributed Denial of Service (DDoS) Attacks in IOT Based Monitoring System of Banking Sector Using Machine Learning Models. *Sustainability*. 2022. Vol. 14, No. 14. 8374. DOI: 10.3390/su14148374

28. Roopak M., Tian G. Y., Chambers J. An Intrusion Detection System Against DDoS Attacks in IoT Networks. *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. Las Vegas, USA, 2020. P. 562–567. DOI: 10.1109/CCWC47524.2020.9031206

29. Atlam H. F., Hemdan E.-D., Alenezi A., Alassafi M. O., Wills G. B. Internet of Things Forensics: A Review. *Internet of Things*. 2020. Vol. 11. 100220. DOI: 10.1016/j.iot.2020.100220

30. Ain N. U. та ін. Securing IoT Networks Against DDoS Attacks: A Hybrid Deep Learning Approach. *Sensors*. 2025. Vol. 25, No. 5. 1346. DOI: 10.3390/s25051346

31. Saiyedand M. F., Al-Anbagi I. Deep ensemble learning with pruning for DDoS attack detection in IoT networks. *IEEE Transactions on Machine Learning in Communications and Networking*. 2024. Vol. 2. P. 596–616. DOI: 10.1109/TMLCN.2024.3395419

32. Wahab S. A. та ін. A Multi-Class Intrusion Detection System for DDoS Attacks in IoT Networks Using Deep Learning and Transformers. *Sensors*. 2025. Vol. 25, No. 15. 4845. DOI: 10.3390/s25154845

33. Fagan M., Megas K. N., Scarfone K., Smith M. Foundational Cybersecurity Activities for IoT Device Manufacturers. Gaithersburg : National Institute of Standards and Technology, 2020. 36 p. DOI: 10.6028/NIST.IR.8259

34. Fagan M., Megas K. N., Scarfone K., Smith M. IoT Device Cybersecurity Capability Core Baseline. Gaithersburg : National Institute of Standards and Technology, 2020. DOI: 10.6028/NIST.IR.8259A

35. Fagan M. та ін. IoT Device Cybersecurity Guidance for the Federal Government: Establishing IoT Device Cybersecurity Requirements. Gaithersburg :

National Institute of Standards and Technology, 2021. DOI: 10.6028/NIST.SP.800-213

36. Guidelines for Securing the Internet of Things : ENISA Report. Athens : European Union Agency for Cybersecurity, 2020. URL: https://www.enisa.europa.eu/sites/default/files/publications/ENISA%20Report%20-%20Guidelines%20for%20Securing%20the%20Internet%20of%20Things.pdf (дата звернення: 23.05.2026).

37. ENISA Threat Landscape 2020: Distributed Denial of Service (DDoS). Athens : European Union Agency for Cybersecurity, 2020. URL: https://www.enisa.europa.eu/sites/default/files/publications/ETL2020%20-%20DDoS%20A4.pdf (дата звернення: 23.05.2026).

38. ENISA Threat Landscape 2020: Botnet. Athens : European Union Agency for Cybersecurity, 2020. URL: https://www.enisa.europa.eu/sites/default/files/publications/ETL2020%20-%20Botnet%20A4.pdf (дата звернення: 23.05.2026).

39. Про затвердження Методичних рекомендацій щодо реагування суб'єктами забезпечення кібербезпеки на різні види подій у кіберпросторі : наказ Адміністрації Держспецзв'язку від 03.07.2023 № 570. *Законодавство України / Верховна Рада України*. URL: https://zakon.rada.gov.ua/go/v0570519-23 (дата звернення: 23.05.2026).

40. Про затвердження Методичних рекомендацій щодо забезпечення кіберзахисту автоматизованих систем управління технологічними процесами : наказ Адміністрації Держспецзв'язку від 29.05.2023 № 463. *Державна служба спеціального зв'язку та захисту інформації України*. URL: https://cip.gov.ua/ua/news/nakaz-administraciyi-derzhspeczv-yazku-vid-29-05-2023-463-pro-zatverdzhennya-metodichnikh-rekomendacii-shodo-zabezpechennya-kiberzakhistu-avtomatizovanikh-sistem-upravlinnya-tekhnologichnimi-procesami (дата звернення: 23.05.2026).

41. Рекомендації. *CERT-UA*. URL: https://cert.gov.ua/recommendations (дата звернення: 23.05.2026).

42. ДЦКЗ Держспецзв'язку запустив нові послуги захисту від кіберзагроз та/або кіберінцидентів. *Державний центр кіберзахисту*. 26.06.2024. URL: https://scpc.gov.ua/uk/articles/368 (дата звернення: 23.05.2026).

43. Україна успішно відбила найбільшу DDoS-атаку в своїй історії. *Державна служба спеціального зв'язку та захисту інформації України*. 16.02.2022. URL: https://cip.gov.ua/ua/news/ukrayina-uspishno-vidbila-naibilshu-ddos-ataku-v-svoyii-istoriyi (дата звернення: 23.05.2026).

44. DDoS threat report for 2024 Q4. *Cloudflare Radar*. 21.01.2025. URL: https://radar.cloudflare.com/reports/ddos-2024-q4 (дата звернення: 23.05.2026).

45. DDoS Threat Intelligence Report. Issue 13: An Era of DDoS Hacktivism. *NETSCOUT*. 2024. URL: https://www.netscout.com/threatreport/1h2024/ (дата звернення: 23.05.2026).

46. DDoS Attack Trends in 2024 Signify That Sophistication Overshadows Size. *Akamai*. 07.04.2025. URL: https://www.akamai.com/blog/security/ddos-attack-trends-2024-signify-sophistication-overshadows-size (дата звернення: 23.05.2026).

47. H1 2024 DDoS Threat Review. *Radware*. 15.08.2024. URL: https://www.radware.com/blog/security/h1-2024-ddos-threat-review/ (дата звернення: 23.05.2026).

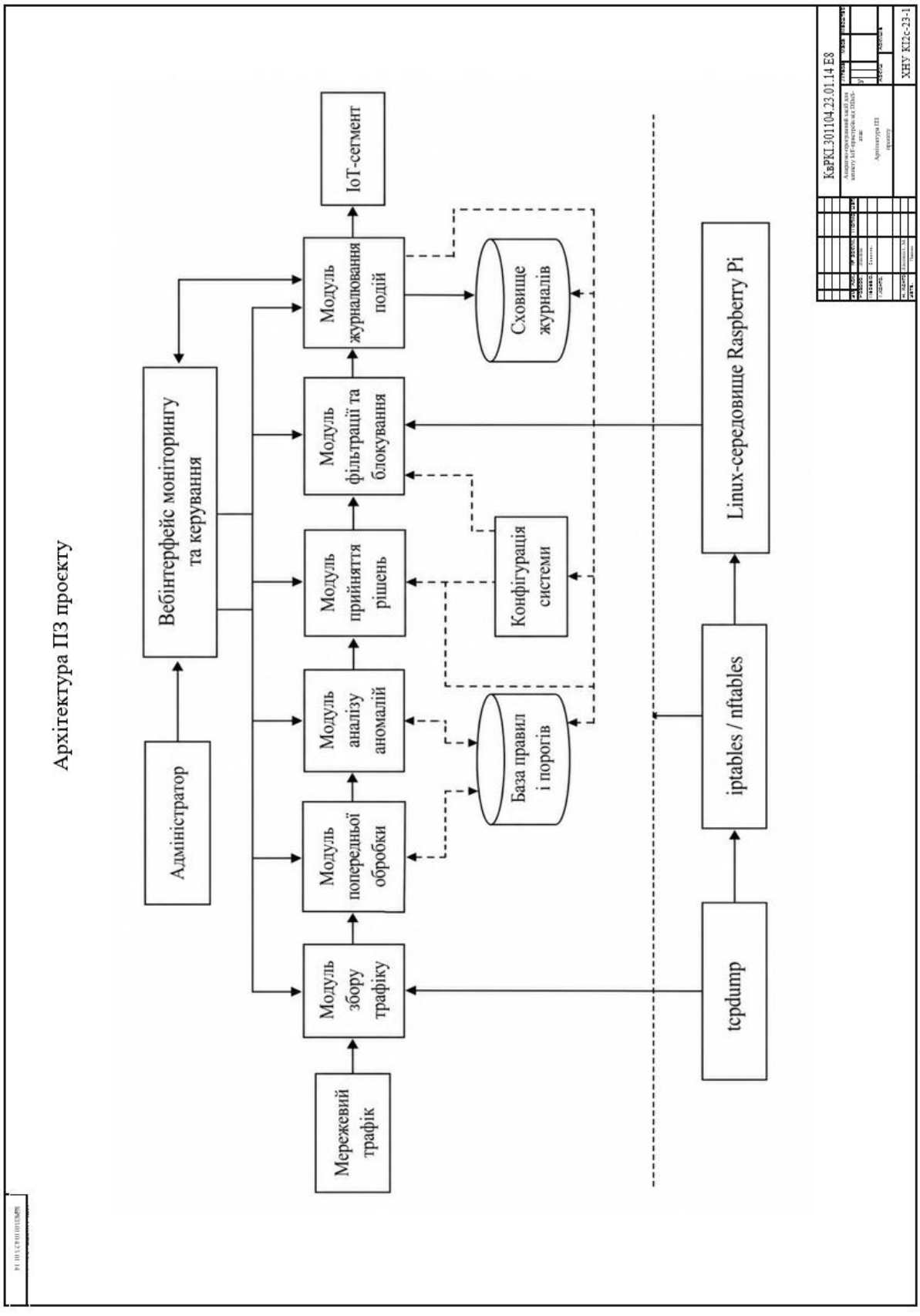
48. Raspberry Pi Documentation. *Raspberry Pi Ltd*. URL: https://www.raspberrypi.com/documentation/ (дата звернення: 23.05.2026).

49. Quick reference-nftables in 10 minutes. *nftables wiki*. 21.04.2024. URL: https://wiki.nftables.org/wiki-nftables/index.php/Quick_reference-nftables_in_10_minutes (дата звернення: 23.05.2026).

ДОДАТОК А

(обов'язковий)

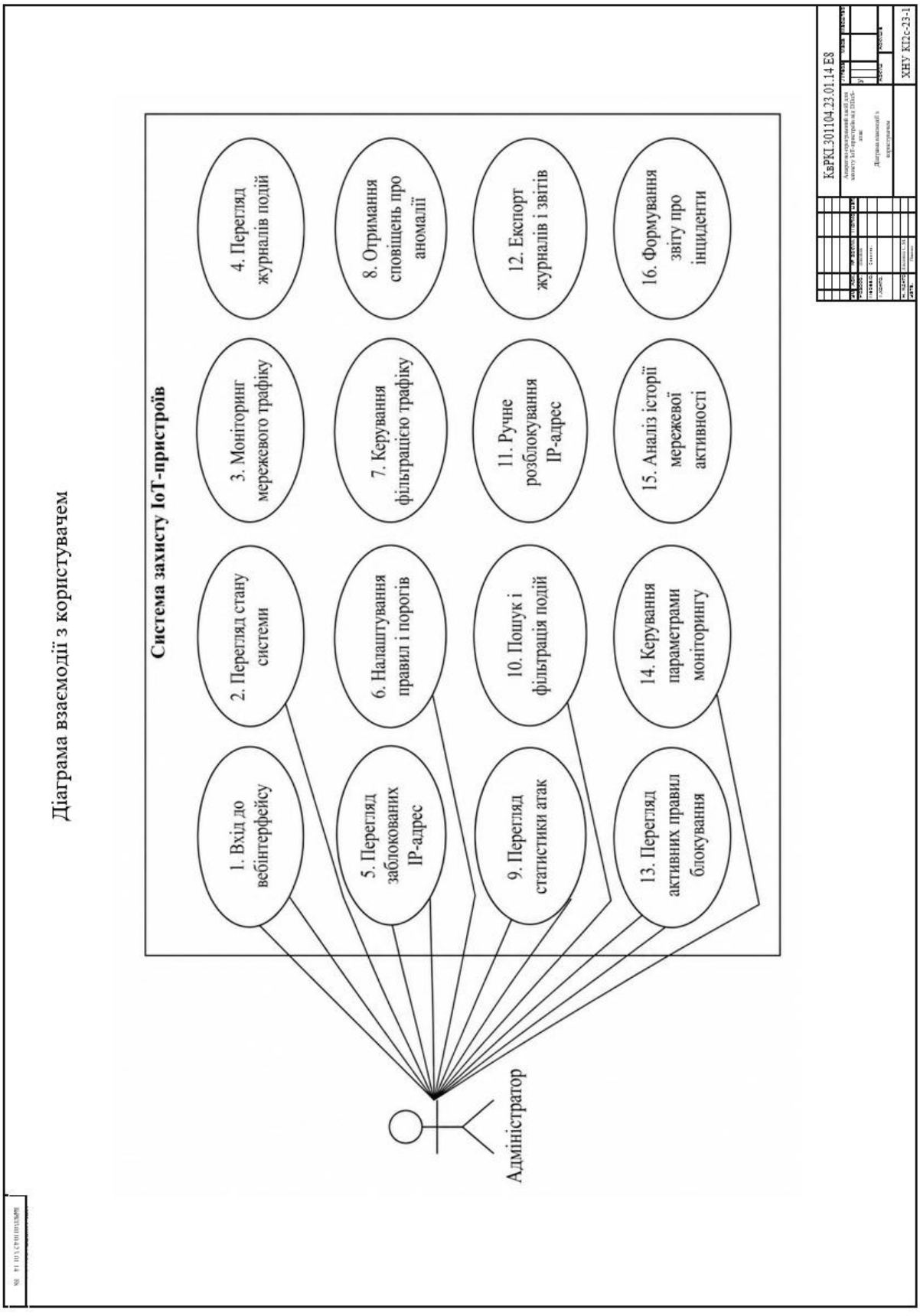
Копія креслення «Архітектура ПЗ проєкту»



ДОДАТОК Б

(обов'язковий)

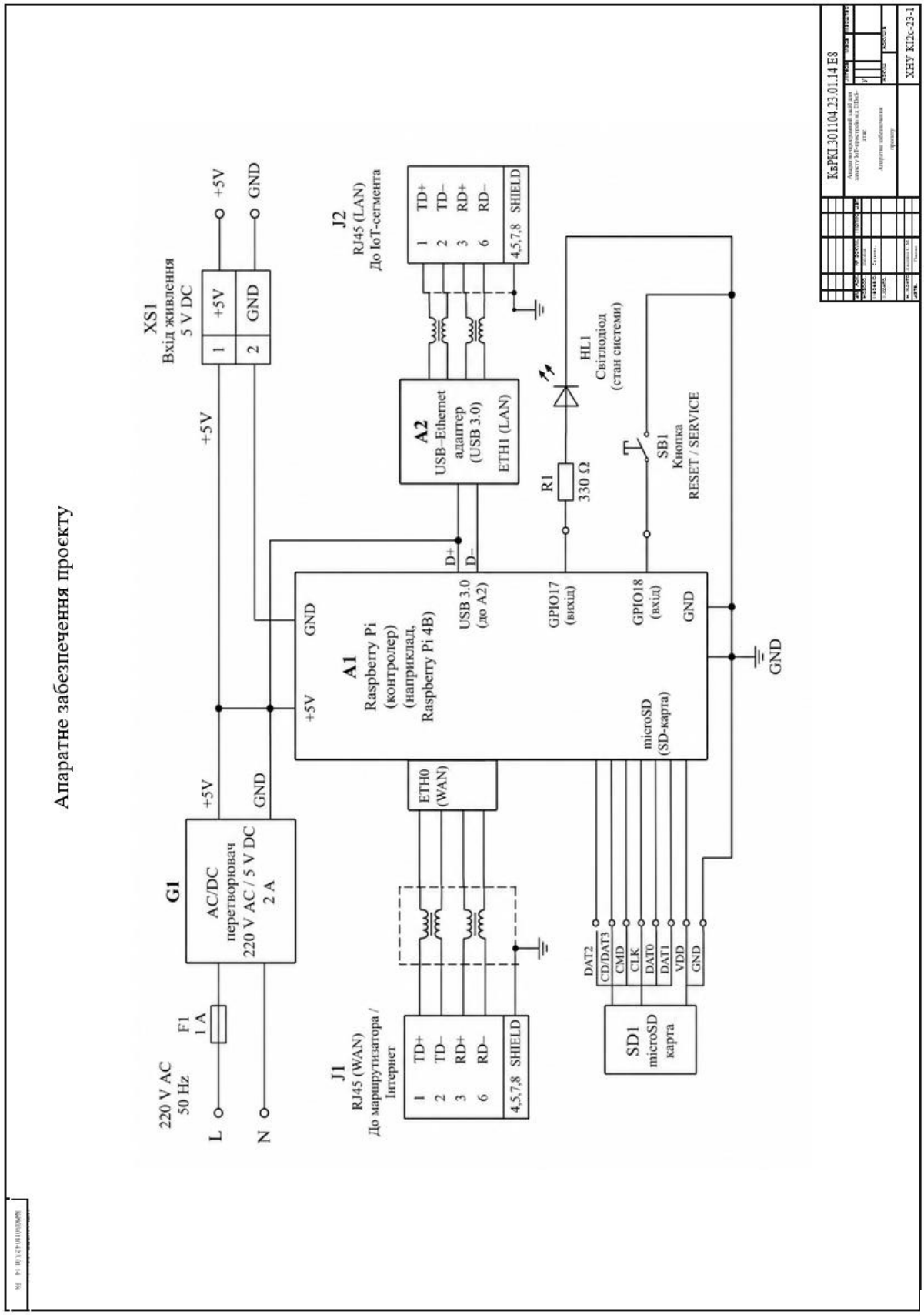
Копія креслення «Діаграма взаємодії з користувачем»



ДОДАТОК В

(обов’язковий)

Копія креслення «Апаратне забезпечення проєкту»



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: Назар ДЗЮБАК

Співавтор:

Назва: Апаратно-програмний засіб для захисту IoT-пристроїв від DDoS-атак

Експерт: Олег САВЕНКО

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:6.55%

Коефіцієнт подібності 2:2.36%

Мікропробіли: 3

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2026-06-06 11:46:23.0

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:

2026-06-06

Дата

Доцент Андрій Нічепорук

експерт

Sat Jun 06 08:50:35 EEST 2026, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (http://ap.km.ua) v-15.701

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 10%**

	Документ		Сумарний збіг по Базі Даних	
ID: 273898 Назва: БКР Апаратно-програмний засіб для захисту IoT-пристроїв від DDoS-атакНазар ДЗЮБАК Додано в БД: 2026-06-06 Автора: Назар ДЗЮБАК Керівники: Олег САВЕНКО Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	118743	913	2016 (2%)	25 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Дзюбак Назар Вікторович

Тема: Апаратно-програмний засіб для захисту IoT-пристроїв від DDoS-атак

Спеціальність**:** 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень ___3____ Кількість сторінок записки ___67___

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є проєктування, реалізація та перевірка функціонування апаратно-програмного засобу для захисту IoT-пристроїв від DDoS-атак на базі Raspberry Pi. Прийняті рішення включають використання одноплатного комп'ютера Raspberry Pi як центрального фільтрувального вузла, розміщеного між основною мережею та захищеним IoT-сегментом, реалізацію програмного модуля збору й аналізу мережевого трафіку, розроблення алгоритму виявлення аномальної активності та механізму автоматичного блокування підозрілих джерел. Для відображення стану системи реалізовано вебінтерфейс моніторингу, у якому подано статистику трафіку, журнал подій безпеки, активні блокування та поточний режим роботи фільтра.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню. Усі вимоги, зазначені у завданні на кваліфікаційну роботу, а саме аналіз предметної області, проєктування апаратно-програмного засобу для захисту IoT-пристроїв від DDoS-атак, розроблення структури мережевого фільтра, реалізація алгоритму виявлення аномального трафіку та створення практичного програмного прототипу, виконані в повному обсязі.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проаналізовано особливості побудови IoT-мереж, загальну характеристику DDoS-атак, ознаки аномального мережевого трафіку та існуючі засоби захисту від перевантаження мережевої інфраструктури. У другому розділі спроєктовано загальну архітектуру апаратно-програмного засобу, обґрунтовано вибір Raspberry Pi

як фільтрувального вузла, визначено структуру програмних компонентів, розроблено алгоритм виявлення аномального трафіку та механізм реагування на DDoS-активність. У третьому розділі виконано практичну реалізацію експериментального стенда, програмного модуля збору й аналізу трафіку, алгоритму виявлення підозрілої активності, механізму автоматичного блокування та вебінтерфейсу моніторингу стану захисту.

4. Позитивні сторони роботи: Висока практична цінність роботи полягає у створенні доступного локального засобу захисту IoT-сегмента, який не потребує дорогого спеціалізованого обладнання та може бути розгорнутий на базі Raspberry Pi. Позитивним є те, що захисну логіку винесено на окремий проміжний вузол, завдяки чому малоресурсні IoT-пристрої не перевантажуються додатковими функціями аналізу та блокування трафіку.

5. Негативні сторони роботи: Суттєвим недоліком є обмежений масштаб практичної перевірки, оскільки працездатність засобу оцінено переважно в умовах лабораторного стенда та змодельованого мережевого навантаження.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно та відповідає вимогам до кваліфікаційних робіт бакалаврського рівня. Структура роботи є логічною, матеріал подано послідовно, а практична частина безпосередньо пов'язана з теоретичними положеннями.

7. Відгук про роботу в цілому: Робота виконана на високому технічному рівні. Здобувач продемонстрував ґрунтовні знання у сфері проєктування сучасних кіберфізичних систем, програмування мікроконтролерів та інтеграції комплексних мережевих рішень.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Яворська Наталія Іванівна, к. пед. наук, доцент кафедри ІПЗ

"18" червня 2026 р. _____ (підпис)

Зав. кафедри КІІС
д-р. філософії Ользі ПАВЛОВІЙ

Назар ДЗЮБАК

ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-23-1

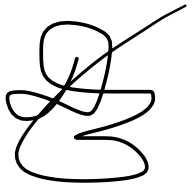
ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

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

1 травня 2026 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Кіберфізична система моніторингу вмісту вуглецю та пилу в повітрі
Автор Назар ДЗЮБАК
Освітня програма Комп'ютерна інженерія та програмування
Рівень вищої освіти перший (бакалаврський)
Спеціальність 123 Комп'ютерна інженерія
Науковий керівник: д.т.н., проф. Олег САВЕНКО

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

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

1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;

2) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;

3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 6.55%; та системою Anti-Plagiarism складає 1.0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

01.06.2026

Завідувач кафедри _____ Підпис Ольга ПАВЛОВА Ім'я, ПРІЗВИЩЕ

Гарант освітньої програми _____ Підпис Андрій НІЧЕПОРУК Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи _____ Підпис Олег САВЕНКО Ім'я, ПРІЗВИЩЕ