Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Заставної Яни Валентинівни

на здобуття ступеня вищої освіти Бакалавра

Система виявлення загроз для IoT-пристроїв на основі поведінкового аналізу

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітня програма Кібербезпека

Шифр КРБКБ.2101118.21.01.07 ПЗ

Виконав студентка 4 курсу група КБ-21-1 Яна ЗАСТАВНА

Керівник канд. техн. наук, доцент ст. Микола СТЕЦЮК

Нормоконтролер старший викладач Сергій МОСТОВИЙ

До захисту допускаю:
Завідувач кафедри кібербезпеки Юрій КЛЬОЦ

9 06 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Рівень вищої освіти Бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ

Завідувач кафедри кібербезпеки

Юрій КЛЬОЦ

15 лютого 2025 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Заставній Яні Валентинівній

1 Тема роботи Система виявлення загроз для IoT-пристроїв на основі поведінкового аналізу

Керівник роботи Микола Стецюк

Затверджено наказом ректора університету від 07 лютого 2025 № 23

2 Строк подання студентом кваліфікаційної роботи на кафедру 1.06.2025

3 Вихідні дані до роботи

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Огляд та аналіз загроз для IoT-пристроїв, Архітектури IoT–пристроїв та їх вразливість, Типи загроз і атак на IoT–пристрої, Методи виявлення та запобігання загрозам, Розробка системи виявлення загроз на основі поведінкового аналізу, Архітектура системи виявлення загроз для IoT–пристроїв, Контролер IoT–мережі, функції системи виявлення загроз, Визначення нормальної поведінки пристроїв в IoT–мережі, Визначення допустимих відхилинь у поведінці пристроїв, Реакція на аномалії та оновлення політик безпеки, Реалізація прототипу системи виявлення аномалій, Вибір середовища розробки та інструментів, Архітектура програмного забезпечення, Реалізація контролерної частини, Реалізація серверної частини, Тестування роботи системи.

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

6 Консультанти розділів кваліфікаційної роботи

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

7 Дата видачі завдання 16 лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Вибір і затвердження теми кваліфікаційної роботи	Лютий	
Ознайомлення з предметною областю	Лютий	
Дослідження існуючих рішень	Лютий	
Постановка задачі	Лютий	
Визначення загальних принципів рішення задачі	Лютий	
Деталізація принципів рішення задачі	Березень	
Впровадження системи виявлення загроз для iot-пристроїв на основі поведінкового аналізу	Березень	
Апробація проєктних рішень	Березень	
Оформлення пояснювальної записки згідно вимог	Травень	
Оформлення графічної частини	Травень	
Захист КР	Червень	

Студентка Яна ЗАСТАВНА

Керівник кваліфікаційної роботи Микола СТЕЦЮК

АНОТАЦІЯ

Тема кваліфікаційної роботи: Система виявлення загроз для IoT-пристроїв на основі поведінкового аналізу.

Автор роботи: Заставна Яна Валентинівна.

Керівник роботи: Стецюк Микола Васильович.

Пояснювальна записка: 67 с., 2 додатки, 12 рисунків, 1 таблиця, 40 джерел.

Графічна частина: 3 плакати, 12 презентаційних слайдів.

АЛГОРИТМ РОБОТИ, ФУНКЦІОНАЛЬНА СХЕМА, ПРОГРАМНИЙ КОД.

Кваліфікаційна робота бакалавра присвячена створенню системи виявлення загроз для пристроїв Інтернету речей (IoT) з використанням методів поведінкового аналізу.

У роботі розглянуто питання забезпечення безпеки IoT-пристроїв, зокрема досліджено типові вектори атак і вразливості, пов'язані з їх використанням у критичних середовищах. Як об'єкт дослідження обрано IoT-термометр, що здійснює передачу даних на комп'ютер у серверній кімнаті. Розроблено підхід до виявлення відхилень у поведінці пристрою на основі аналізу мережевого трафіку.

У ході дослідження проаналізовано особливості поведінкових моделей, обґрунтовано вибір методу аналізу, сформульовано загальні вимоги до системи виявлення загроз та визначено напрями її подальшого розвитку.

02.06.2025

ABSTRACT

Subject of qualification work: Threat detection system for IoT devices based on behavioral analysis.

Author: Zastavna Yana Valentinivna.

Head of work: Stetsyuk Mykola Vasilyevich.

Explanatory note: 67 p., 2 appendices, 12 figures, 1 table, 40 sources.

Graphic part: 3 posters, 12 presentation slides.

ALGORITHM OF WORK, FUNCTIONAL DIAGRAM, PROGRAM CODE.

The bachelor's qualification work is devoted to the creation of a threat detection system for Internet of Things (IoT) devices using behavioral analysis methods.

The work considers the issue of ensuring the security of IoT devices, in particular, typical attack vectors and vulnerabilities associated with their use in critical environments are investigated. An IoT thermometer that transmits data to a computer in a server room is chosen as the object of research. An approach to detecting deviations in the behavior of the device based on network traffic analysis is developed.

During the study, the features of behavioral models are analyzed, the choice of the analysis method is justified, general requirements for the threat detection system are formulated, and directions for its further development are determined.

02.06.2025

ЗМІСТ

Вступ 7
1 Огляд та аналіз загроз для IoT–пристроїв 9
1.1 Архітектури IoT–пристроїв та їх вразливість 9
1.2 Типи загроз і атак на IoT–пристрої 15
1.3 Методи виявлення та запобігання загрозам 22
2 Розробка системи виявлення загроз на основі поведінкового аналізу 32
2.1 Архітектура системи виявлення загроз для IoT–пристроїв 32
2.2 Контролер IoT–мережі, функції системи виявлення загроз 36
2.3 Визначення нормальної поведінки пристроїв в IoT–мережі 39
2.4 Визначення допустимих відхилинь у поведінці пристроїв 40
2.5 Реакція на аномалії та оновлення політик безпеки 43
3 Реалізація прототипу системи виявлення аномалій 48
3.1 Вибір середовища розробки та інструментів 48
3.2 Архітектура програмного забезпечення 50
3.3 Реалізація контролерної частини 52
3.4 Реалізація серверної частини 54
3.5 Тестування роботи системи 56
Висновки 60
Перелік джерел посилання 63
Додаток А 68
Додаток Б 71

ВСТУП

IoT швидко змінює спосіб взаємодії людей з технікою, пропонуючи великий обсяг застосувань, від «розумних» будинків до промислових систем та медичних приладів. Паралельно з розвитком інтернету та гаджетів також зросла і кількість кіберзагроз, пов'язаних з їхньою роботою. Низький рівень безпеки на пристроях, незахищенні мережі роблять IoT–пристрої легкою мішенню до різних видів кібератак, таких як DDoS–атаки, експлойт програмного забезпечення.

Особливості IoT–інфраструктури створюють нові можливості у забезпеченні її від кібератак. На відміну від традиційних IT–систем, IoT–пристрої дають можливість покращувати скорочуючи час виконання і потреби в ресурсах. Традиційні методи захисту IoT–систем, такі як статичні правила та сигнатури частіше за все виявляються неефективними перед новим атаками і зловмисники створюють нові прийоми атак, які вже важче виявити за допомогою стандартних правил. У таких умовах найкращим рішенням є впровадження систем виявлення загроз, що базуються на поведінковому аналізі. Цей метод дозволяє здійснювати моніторинг і аналіз поведінки Інтернету речей у реальному часі, щоб виявити аномалії, які можуть свідчити про атаку. Ці сервіси не лише забезпечують реєстрацію, а й допомагають виявляти нові загрози та оцінювати безпеку.

Метою даної роботи є розробка системи виявлення загроз IoT–пристроїв на основі поведінкового аналізу, забезпечення вчасного виявлення та реагування на аномалії або підозрілі дії в мережах Інтернету речей, що можуть вказувати на порушення безпеки або наявність кіберзагроз. Система повинна ефективно проводити аналіз поведінки IoT–пристроїв, що базуються на основі даних про їх активність, стан та взаємодію з іншими пристроями або ж системами, виявляючи аномалію. Ключовим завданням є забезпечення високого рівня безпеки в IoT–пристроях без необхідного попереднього знання про конкретний тип атаки. Це можна зробити за допомогою поведінкового аналізу, який допомагає виявляти аномалії у реальному часі, що можуть бути ознакою різних загроз або некоректної

роботи. Поведінковий аналіз представляє процес вивчення і аналіз поведінки пристроїв або ж користувачів у системі з метою виявлення аномалій у роботі. Збір даних про активність пристрою або пристроїв є фундаментом для побудови профілю нормальної роботи гаджетів. Це може включати в себе сукупність критеріїв, по типу підключення до мережі, енергоспоживання, час роботи та інші особливості роботи пристрою при нормальній активності.

Аномалії у роботі пристроїв можуть різнитися від підвищеного енергоспоживання до змін у структурі мережевого трафіку, які вказують на можливу атаку. Якщо виявлено підозрілу поведінку, система може генерувати сигнал тривоги і негайно реагує на потенційні загрози.

Щоб система виявлення загроз була ефективнішою, вона повинна бути інтегрованою в загальну архітектуру безпеки мережі. Це означає, що поведінкова аналітика повинна взаємодіяти з іншими системами безпеки, такими як брандмауер, системи виявлення вторгнень (IDS) та інструменти моніторингу мережевого трафіку. Така інтеграція створює єдину платформу для моніторингу, аналізу та реагування загрози, що значно підвищує ефективність захисту.

Загалом, розробка таких систем виявлення загроз є важливим кроком на шляху до підвищення безпеки мереж Інтернету речей, дозволяючи їм своєчасно реагувати на зростаючі кіберзагрози та забезпечувати надійний захист користувачів і організацій, що використовують ці технології.

1 ОГЛЯД ТА АНАЛІЗ ЗАГРОЗ ДЛЯ ІОТ–ПРИСТРОЇВ ТА МЕТОДІВ ЇХ ВИЯВЛЕННЯ

1.1 Архітектури IoT–пристроїв та їх вразливості

Архітектура пристроїв Інтернету речей (IoT) – це багаторівнева структурна основа, яка забезпечує ефективну взаємодію великої кількості пристроїв, здатних до обміну даними в режимі реального часу. Вона поєднує в собі апаратні складові, програмне забезпечення та мережеві протоколи, формуючи екосистему, яка надає широкий спектр функціональних можливостей. Однак, ця ж складність зумовлює виникнення низки викликів, зокрема у сфері безпеки, оскільки велика кількість пристроїв з різноманітними можливостями та стандартами перетворюється на потенційні мішені для кібератак. Відсутність узгодженого підходу до забезпечення безпеки лише загострює ці загрози.

Архітектура IoT складається з чотирьох рівнів, які тісно взаємодіють між собою: рівень пристроїв, мережевий рівень, рівень управління та прикладний рівень. Кожен із цих рівнів має свої унікальні функції, що забезпечують загальну функціональність і надійність системи. Завдяки такій структурі можна проектувати, впроваджувати та масштабувати системи IoT для різних галузей, включно з промисловістю, медициною, транспортом, сільським господарством та побутовою автоматизацією [1].

Рівень пристроїв відкриває багаторівневу архітектуру IoT. Саме тут відбувається фізичний контакт системи IoT з оточенням. Пристрої, що формують цей рівень – це сенсори, виконавчі механізми, камери, мікрофони та інші, що здатні «відчувати» навколишній світ, фіксуючи дані. Вони розроблені для вимірювання широкого спектра параметрів: температура, вологість, рух, тиск, освітлення, рівень забруднення та інше. Наприклад, у сільському господарстві датчики вологості ґрунту допомагають фермерам оптимізувати використання води, зменшуючи витрати та покращуючи врожайність. У «розумних» містах датчики руху регулюють роботу світлофорів, забезпечуючи оптимальний потік

транспорту. У медицині пристрої цього рівня збирають життєво важливі показники пацієнтів, такі як серцевий ритм чи рівень цукру в крові, і передають їх для аналізу. Головна мета цього рівня – збір інформації, що далі надсилається для аналізу на наступний рівень. Але з огляду на обмежені можливості обладнання (мале споживання енергії, незначні обчислювальні ресурси) виникають певні труднощі. Це породжує нові вимоги до розробки систем, бо необхідно досягти рівноваги між функціональністю, ефективним використанням енергії та безпекою.

Мережевий рівень, або транспортний рівень, виконує критично важливу функцію передачі даних від пристроїв першого рівня до рівня управління. Він використовує різноманітні технології зв'язку, такі як Wi–Fi, Bluetooth, Zigbee, LoRaWAN, стільникові мережі (наприклад, 4G, 5G) та супутниковий зв'язок [2].

Наприклад, в розумному будинку дані з температурних сенсорів можуть надходити через локальну мережу Wi–Fi до центрального вузла, який, у свою чергу, керує термостатом. У промислових умовах застосовують протоколи з низьким енергоспоживанням, як–от LoRaWAN, щоб гарантувати передавання інформації з датчиків на значні відстані. Основною вимогою до мережевого рівня є забезпечення безперебійної передачі даних у реальному часі. При цьому важливими критеріями залишаються швидкість передачі, стабільність, енергоефективність та захищеність. Незахищені мережі можуть стати точкою входу для зловмисників, що використовують їх для перехоплення даних або компрометації системи. Саме тому впровадження протоколів безпеки, таких як шифрування переданих даних, є обов'язковим на цьому рівні [3].

На управлінському рівні, котрий інколи означає як етап опрацювання. Зібрані з первинних пристроїв відомості зазнають обробки, зберігання та ретельного аналізу. Саме тут в дію вступають сервери, бази даних та хмарні платформи, що надають гнучкість у масштабуванні та високу ефективність. Хмарні служби, як–от Amazon Web Services (AWS) чи Microsoft Azure, пропонують потужний набір інструментів для збереження даних, їх детального

вивчення та складання відповідних звітів. Окрім обробки даних, цей рівень також відповідає за управління пристроями в системі IoT. Наприклад, у «розумному» будинку цей рівень може автоматично оновлювати програмне забезпечення пристроїв або оптимізувати їхню роботу [4].

Прикладний рівень – це кінцева ланка архітектури IoT. Він надає змогу користувачам взаємодіяти із системою через мобільні або ж веб–застосунки. На цьому рівні накопичені та оброблені дані служать для розробки корисних сервісів. Скажімо, у «розумному» помешканні користувач має змогу керувати роботою пристроїв через застосунок на смартфоні. У промисловості цей щабель забезпечує моніторинг обладнання, убезпечуючи від аварійних випадків. Завдяки цьому рівню IoT–системи стають доступними та зручними для кінцевих користувачів, відкриваючи можливості для впровадження нових сервісів і бізнес–моделей [5].

Пристроям Інтернету речей часто бракує необхідного захисту для протидії загрозам безпеки. Поширені вразливості та ризики дозволяють кіберзлочинцям компрометувати пристрої та використовувати їх як плацдарм для сучасних кібератак. Обмежена обчислювальна здатність пристроїв Інтернету речей зумовлює малу кількість варіантів для надійного захисту даних та гарантування безпеки, необхідних для відсічі кібернападів. Використання різноманітних технологій передачі даних цими пристроями ускладнює розробку та впровадження відповідних методів і протоколів безпеки [6].

Пристрої Інтернету речей можуть бути скомпрометовані через різні вразливості:

– слабкі чи надто прості паролі – один із найчастіших прийомів, що його застосовують зловмисники для зламу гаджетів Інтернету речей. Зловмисники легко можуть підібрати прості та такі, що повторюються паролі, котрі або дуже короткі, або їх легко вгадати, аби використати їх для захоплення контролю над пристроями та організації великомасштабних атак [7];

– у незахищеній мережі кіберзлочинці можуть легко використовувати слабкі місця в протоколах і сервісах, що працюють на пристроях. Після того, як

мережа використовується, зловмисники можуть отримати доступ до конфіденційних даних, якими обмінюються користувацькі пристрої та сервери. Незахищені мережі особливо вразливі до атак типу «людина посередині» (MITM), метою яких є викрадення облікових даних та аутентифікації пристроїв в рамках більш масштабної кібератаки [8];

– вразливі інтерфейси екосистеми, зокрема інтерфейси прикладного програмування (API), мобільні та веб–додатки, створюють можливості для зловмисників компрометувати пристрої. Організаціям критично важливо впроваджувати процедури аутентифікації та авторизації, що забезпечують перевірку автентичності користувачів і захист хмарних і мобільних інтерфейсів. Практичні інструменти аутентифікації здатні допомогти серверам диференціювати легітимні пристрої від тих, що використовуються зловмисниками [9];

– пристрої з незахищеними процесами оновлення піддаються ризику встановлення шкідливого або несанкціонованого коду, прошивки або програмного забезпечення. Пошкоджені оновлення можуть скомпрометувати пристрої, що може бути критично важливим для енергетичних, медичних та промислових організацій. Оновлення повинні бути безпечними та зашифрованими, а все програмне забезпечення має бути протестованим та перевіреним [10];

– екосистема Інтернету речей вразлива через слабкі місця в коді та програмному забезпеченні, плюс старі системи. Застосування незахищених або давно оновлених частин, наприклад, відкритого коду чи стороннього програмного забезпечення, може створити умови для проникнення, що збільшують ризики атак на організацію [11].

Користувачі повинні повною мірою усвідомлювати ризики, що виникають унаслідок підключення пристроїв до Інтернету речей (IoT), і вживати відповідних заходів для забезпечення їхньої безпеки. Усі пристрої, які мають доступ до Інтернету, можуть стати потенційною мішенню для кіберзлочинців, якщо не

вжити необхідних заходів для їх захисту. Для того щоб зменшити можливі загрози, користувачі повинні не тільки розуміти ці ризики, але й активно працювати над їх нейтралізацією.

Одним з найважливіших завдань є аранжування пріоритетів безпеки для підключених пристроїв. Кожен прилад, що під'єднаний до домашньої чи офісної мережі, виступає потенційним місцем проникнення для зловмисників. Відтак, критично важливо заздалегідь оцінити всі можливі загрози для кожного пристрою та вжити відповідних заходів для їх усунення. Одним із першочергових кроків, які мають зробити користувачі, є зміна стандартних паролів для IoT–пристроїв. Багато з них надходять з заводськими паролями, котрі легко доступні в Інтернеті, а тому їх слід негайно замінювати на більш комплексні та унікальні. Окрім зміни паролів, важливим кроком у захисті пристроїв є регулярне оновлення програмного забезпечення та прошивки. Існують численні випадки, коли зловмисники використовують відомі вразливості в старих версіях прошивки для атак на пристрої. Регулярні оновлення допомагають захистити пристрої від нових загроз, оскільки вони містять патчі для виявлених уразливостей. У багатьох сучасних пристроях можна налаштувати автоматичне оновлення, що значно спрощує процес підтримки пристроїв у безпечному стані [11].

Ще одним надзвичайно важливим етапом є впровадження відповідних налаштувань безпеки для пристроїв, які взаємодіють з мережею. Це здатне охоплювати деактивацію певних функцій, здатних створити загрозу для безпеки, як–от віддалений доступ до пристрою через інтернет або ненадійні порти, котрі можуть використовуватися для кібератак. Разом з тим, користувачам наполегливо радять використовувати міцні паролі та, за наявності опції, активувати двофакторну автентифікацію. Це дозволить сформувати додатковий щабель захисту, навіть якщо пароль буде розшифровано [12].

Особливо важливо забезпечити безпеку роутерів, які є основними пристроями для підключення до Інтернету в будь–якій мережі. Роутери часто є точкою доступу для хакерів, і саме через них можуть бути здійснені атаки на інші

пристрої, підключені до мережі. Для захисту роутера необхідно змінити пароль за замовчуванням, налаштувати шифрування Wi–Fi на високий рівень (наприклад, WPA3), а також відключити небажані функції, такі як WPS, які можуть бути вразливими для атак [13].

На додаток до цього, користувачі зобов'язані активно докладати зусиль до безпеки своїх мереж, дотримуючись принципів безпеки мережевої інфраструктури. Ключовим аспектом є застосування схем шифрування для захисту даних, що передаються через мережу. Скажімо, шифрування даних за допомогою протоколів SSL/TLS дає змогу захистити інформацію під час її переміщення в Інтернеті, що значно ускладнює доступ до неї зловмисникам. До того ж, в організаціях необхідно впроваджувати інфраструктуру відкритих ключів (PKI), котра дозволяє забезпечити надійну аутентифікацію пристроїв та користувачів, а також їх захист від підробок.

Один із важливих аспектів захисту в організаціях – це постійний моніторинг мережевої активності. Використання інструментів для моніторингу трафіку дозволяє виявляти будь–яку незвичну або потенційно шкідливу активність, яка може вказувати на спробу проникнення в мережу. Такими інструментами можуть бути системи виявлення та запобігання вторгнень (IDS/IPS), які аналізують поведінку мережі та виявляють аномалії [14]. Регулярні перевірки і сканування вразливостей також є необхідними для забезпечення безпеки пристроїв, підключених до мережі IoT. Існують також спеціальні інструменти для сканування вразливостей IoT, які дозволяють перевіряти пристрої на наявність відомих вразливостей або помилкових налаштувань безпеки. Ці інструменти дозволяють виявити потенційні проблеми та допомогти користувачам та організаціям вчасно вжити заходів для їх усунення, до того як вони будуть використані зловмисниками. Застосування подібних сканерів є важливою частиною комплексного підходу до забезпечення безпеки пристроїв IoT. Для досягнення максимального рівня безпеки важливо не тільки налаштувати окремі пристрої, але й забезпечити комплексну безпеку всієї інфраструктури IoT. Це

включає не лише захист фізичних пристроїв, але й забезпечення належного рівня захисту даних, які обробляються та передаються між пристроями. Застосування принципів безпеки на кожному етапі життєвого циклу IoT–пристроїв допоможе створити надійну систему, що зменшує ризики і мінімізує вплив потенційних загроз.

Як наслідок, користувачі та організації повинні враховувати всі можливі загрози та активно вживати заходів для забезпечення захисту своїх IoT–пристроїв. Надзвичайно важливим є комплексний підхід до безпеки, який охоплює технічні рішення, налаштування та навчання користувачів. Підвищення обізнаності, періодичні оновлення та впровадження сучасних методів захисту дадуть змогу мінімізувати ризики та гарантувати належний рівень захисту для пристроїв, підключених до Інтернету речей [15].

1.2 Типи загроз і атак на IoT–пристроїв

Можливість зловмисників отримати доступ до конфіденційних даних, що зберігаються на пристроях IoT, є серйозною загрозою для безпеки. До таких даних належать персональні відомості, зокрема імена, адреси, паролі, а також чутлива фінансова, корпоративна чи навіть військова інформація. Під ризиком опиняються і користувачі приватних мереж, і державні установи. Несанкціоновані або неперевірені пристрої, інтегровані в корпоративну чи державну інфраструктуру, можуть стати каналом для несанкціонованого проникнення в інші частини системи. Такі пристрої часто використовуються для шпигунства, викрадення конфіденційних даних або створення умов для подальших атак.

Водночас одним з основних викликів залишається нехтування безпекою під час інтеграції IoT–пристроїв у розгалужені мережі. Скажімо, підключення пристроїв, які ще мають заводські налаштування, без заміни стандартних паролів

або застосування відповідного шифрування, несе в собі серйозну загрозу. Кібер–злочинці можуть без проблем знаходити такі пристрої за допомогою мережевих сканерів і здобувати до них доступ [16].

У випадку з корпоративними або державними мережами, компрометація одного пристрою може призвести до ланцюгової реакції, відкриваючи доступ до всіх інших підключених систем. Зловмисники можуть не лише викрасти інформацію, але й змінити роботу системи або використовувати пристрої для дестабілізації її функціонування. Наприклад, у промисловості компрометація IoT–пристроїв може спричинити зупинку виробничих процесів, а у військовій сфері – розголошення стратегічних даних.

Крім того, відчутною проблемою є слабкий нагляд за виникненням нелегальних пристроїв в мережах. Під'єднання навіть одного пристрою без належної перевірки може послужити точкою входу для атак «людина посередині» (MITM) або застосування бот–мереж задля масштабних DDoS–атак. Щоб мінімізувати ці ризики, необхідно впроваджувати багаторівневу систему захисту, яка включає автентифікацію пристроїв, шифрування переданих даних, регулярний моніторинг мережі на предмет підозрілої активності та обмеження доступу для несанкціонованих пристроїв. У державному та корпоративному секторах також варто розробляти політики безпеки для IoT, включаючи обов'язкове тестування пристроїв перед підключенням до мережі, впровадження систем раннього попередження про можливі загрози та регулярне оновлення програмного забезпечення [17].

Загрози безпеці IoT бувають різних форм, від простого злому паролів до більш складних атак, які використовують вразливості в пристроях IoT (рис. 1.1).

Спуфінг, або ж підміна – це кібернаступ, коли зловмисники вдають з себе перевірене джерело для здобуття доступу до відомостей чи секретних даних. Імітування іншої особи можливе через інтернет–сторінки, email, телефонні з'єднання, повідомлення в месенджерах, IP–адреси, сервери та інше. Зазвичай завдання спуфінгу полягає у отриманні доступу до персональної інформації,

викраденні фінансів, поширенні шкідливого програмного забезпечення. Існує чимало різновидів такого нападу: підміна номеру абонента, підробка веб–сайту, фальсифікація email, спуфінг ip, підміна dns–сервера, arp спуфінг, спуфінг смс, gps спуфінг, атака «людина посередині», підробка форматів [18].

Рисунок 1.1 – Загрози безпеки в IoT

Загрози протоколу NDP (Neighbor Discovery Protocol) – це протокол мережевого рівня, який використовується в мережах IPv6 для вирішення різних завдань, пов'язаних з мережевими пристроями, таких як генерація IP–адрес, пошук маршрутів і дозвіл адрес для пристроїв в мережі. Подібно до IPv4 ARP, NDP можна використовувати для виявлення MAC–адрес інших пристроїв у мережі на основі адреси IPv6. Протокол дозволяє виявляти маршрутизатори в мережі шляхом трансляції рекламних повідомлень маршрутизатора на мережевому інтерфейсі. Рекламні повідомлення маршрутизаторів використовуються для налаштування мережевої конфігурації, наприклад, адреси підмережі, шляхом розподілу префіксів між маршрутизаторами. NDP може використовуватися для виявлення, коли сусідній пристрій більше не доступний, і

надсилання повідомлень про недоступність. Протокол дозволяє пристроям автоматично отримувати IPv6–адреси з мережі без використання протоколу динамічної конфігурації хоста (DHCP) [19].

Атаки на пристрої Інтернету речей (IoT) – це дії з боку зловмисників, спрямовані на використання слабких місць у пристроях, що мають доступ до Інтернету. До таких пристроїв належать: «розумні» побутові прилади, системи керування промисловими об’єктами, медичне обладнання тощо. Зловмисники прагнуть отримати контроль над цими пристроями, щоб викрасти конфіденційну інформацію або використати пристрої в ботнетах для реалізації інших шкідливих цілей [20].

Розподілені атаки на відмову в обслуговуванні (DDoS–атаки) спрямовані на веб–сайти та сервери, щоб порушити роботу мережевих сервісів. Метою таких атак є споживання ресурсів додатків. Зловмисники заливають веб–сайти фальшивим трафіком, що призводить до повільної роботи або повного закриття веб–сайту. Поширеність таких атак зростає; від DDoS–атак страждає велика кількість компаній, що працюють у різних галузях. Деякі сектори, такі як ігрова індустрія, електронна комерція та телекомунікації, піддаються більшому ризику, ніж інші. DDoS–атаки є однією з найпоширеніших кіберзагроз, які ставлять під загрозу безпеку бізнесу, веб–сайтів, продажів та репутації. Під час DDoS–атаки кілька ботів або цілих бот–мереж перевантажують веб–сайт або сервіс HTTP–запитами та трафіком. По суті, кілька комп'ютерів намагаються захопити один пристрій, роблячи його недоступним для реальних користувачів. Це може спричинити затримки в роботі сервісів та інші збої. Під час атаки хакери також можуть зламати бази даних і отримати доступ до конфіденційної інформації. DDoS–атаки використовують вразливості і націлені на будь–яку загальнодоступну кінцеву точку в Інтернеті. Атаки на відмову в обслуговуванні можуть тривати годинами або навіть днями. Такі атаки можуть спричинити одразу кілька небезпек. Вони можуть загрожувати особистим і професійним пристроям [21].

Зловмисники здатні використовувати ненадійні паролі та інші слабкі місця системи, щоб отримати доступ до пристроїв Інтернету речей, перехопити над ними контроль або вкрасти особисті дані користувачів. Це може стосуватися не лише можливості підглядати через камеру чи прослуховувати мікрофон пристрою, але й використання IoT–приладів як частини бот–мережі для масштабних DDoS–атак, які блокують доступ до критичних сервісів та ресурсів.

Іноді хакери можуть використовувати навіть малопомітні вразливості, які здаються незначними, але можуть мати серйозні наслідки. Наприклад, пристрій може бути інфікований шкідливим програмним забезпеченням, яке дозволяє зловмисникам не тільки отримати доступ до особистих даних, але й здійснювати маніпуляції з іншими пристроями в мережі, що значно підвищує рівень загрози. У разі використання таких вразливих пристроїв для атак на сервери або інші критичні системи, може статися збій у роботі компанії чи державної установи, а також викрадення чутливої інформації.

Крім того, зловмисники здатні застосовувати IoT–пристрої для розгортання шпигунства за користувачами. Зокрема, йдеться про запис особистих бесід або відстеження їхніх дій за допомогою інтернет–під'єднаних камер відеоспостереження та мікрофонів. Подібні дії можуть бути замаскованими та тривати доволі довго, допоки користувач не помітить порушення власної приватності. Ці загрози ще більше ускладнюються тим, що без належного захисту пристрої IoT можуть самі по собі виступати в якості «вхідної точки» для більш масштабних атак. В разі проникнення зловмисників до одного пристрою, це може створити можливості для поширення атак на всю мережу, де всі пристрої стануть потенційно вразливими [22].

Перехоплення сесії (Session Hijacking) – це різновид кібернетичної атаки, спрямованої на несанкціоноване проникнення до сесії користувача в веб–застосунку або мережевому сервісі. Зловмисники реалізують це шляхом перехоплення або крадіжки токена сесії користувача (зазвичай це невеликий фрагмент даних або текстовий рядок, що ідентифікує сесію користувача). Маючи

контроль над сесією, зловмисник здатен здійснювати операції на зразок оформлення покупок від імені користувача, отримання доступу до конфіденційних даних або модифікації налаштувань облікового запису. Цей вид атаки особливо критичний, коли користувач знаходиться в безпечній сесії, приміром, у системі онлайн–банкінгу, оскільки зловмисник може отримати доступ до фінансових даних. Методи, які можуть застосовувати зловмисники для перехоплення сесії користувача, включають в себе перехоплення пакетів, міжсайтовий скриптинг (XSS) та атаки типу «людина посередині». Щоб запобігти перехопленню сесії, розробники веб–додатків та мережеві адміністратори можуть використовувати різноманітні заходи безпеки, наприклад, безпечні cookie–файли, шифрування трафіку та часту ротацію токенів сесії. Додатково, користувачі можуть підвищити рівень власної безпеки, уникаючи незахищених публічних Wi–Fi мереж та застосовуючи двофакторну автентифікацію, де це передбачено [23].

Фізичне втручання є однією з найсерйозніших загроз для пристроїв Інтернету речей (IoT). Зловмисник, який отримав фізичний доступ до пристрою, може здійснити широкий спектр шкідливих дій, таких як викрадення конфіденційних даних, встановлення шкідливого програмного забезпечення або навіть модифікація внутрішньої схеми пристрою. Це може включати підключення до апаратних портів, зчитування даних із внутрішніх накопичувачів або навіть фізичне видалення компонентів для подальшого аналізу. Такий доступ дозволяє зловмисникам проникати в мережу і отримувати доступ до інших підключених пристроїв, створюючи ланцюгову реакцію порушення безпеки.

Ненадійні сполучення між серверами та приладами IoT являють собою ще один серйозний виклик. Зловмисники можуть перехоплювати мережевий трафік, якщо він не достатньо захищений, вдаючись до таких технік, як атаки «людина посередині» (MITM). Це надає їм можливість доступу до секретної інформації, на кшталт паролів, фінансових даних або навіть особистої інформації користувачів. Особливо вразливі пристрої з мікрофонами та камерами, оскільки зловмисники

здатні використовувати їх для прослуховування розмов або отримання візуальної інформації, що може завдати серйозної шкоди приватності [24].

Парольні атаки залишаються одним із найпоширеніших способів злому пристроїв IoT. Багато пристроїв постачаються зі стандартними або слабкими паролями, які легко вгадати або зламати. Зловмисники використовують методи, такі як перебір паролів (brute–force) або словникові атаки, щоб підібрати правильну комбінацію. Крім того, деякі пристрої не підтримують складні паролі або багатофакторну аутентифікацію, що значно полегшує завдання зловмисників. Такі недоліки, спрямовані на забезпечення зручності для користувачів, фактично створюють серйозні ризики безпеки [25].

Операційні системи IoT–пристроїв нерідко містять уразливості, які стають мішенню для кіберзлочинців. Це можуть бути недопрацювання в коді, незалатані дірки або помилки, що залишилися після того, як пристрій вийшов на ринок. Зловмисники активно шукають такі слабкі місця в пристроях, щоб отримати перший доступ. Після цього вони можуть вдатися до ескалації привілеїв, підвищуючи свій рівень до адміністраторського. Це дає їм змогу отримати доступ до важливих даних, змінювати систему або ж повністю заволодіти пристроєм.

Окрім безпосередніх атак, зловмисники також можуть експлуатувати відсутність оновлень безпеки в пристроях IoT. Часто виробники або не надають регулярних оновлень, або користувачі ігнорують їх установку, що створює значні ризики. Відсутність належного шифрування трафіку та аутентифікації між пристроями і серверами також є серйозною проблемою. Наприклад, дані, що передаються у незашифрованому вигляді, можуть бути легко перехоплені, а недосконалі протоколи аутентифікації дозволяють зловмисникам видавати себе за легітимних користувачів або пристрої.

Ще одним чинником, що сприяє атакам, виступає комплексність та взаємозалежність теперішніх IoT–екосистем. За великого відсотка випадків, IoT–пристрої здійснюють обмін інформацією з іншими пристроями або серверами у реальному часі. Якщо один з пристроїв опинився скомпрометованим, це може

потягнути за собою наслідки для усієї мережі. Наприклад, зловмисник може використовувати вразливий пристрій як «вхідну точку» для проникнення до головної мережі, крадіжки даних або розгортання подальших атак на інші підключені системи.

Крім технічних вразливостей, не менш важливим є людський фактор. Користувачі часто не усвідомлюють потенційних ризиків або не дотримуються найкращих практик із безпеки, таких як регулярна зміна паролів, оновлення прошивки або використання функцій двофакторної аутентифікації. Це створює додаткові можливості для зловмисників, які можуть скористатися необізнаністю користувачів або їх недбалим ставленням до безпеки.

Задля мінімізації загроз та забезпечення безпеки пристроїв IoT, ключовим є всебічний підхід до захисту. Він передбачає фізичний захист самих пристроїв, впровадження найсучасніших методів шифрування даних, обов'язкове та регулярне оновлення програмного забезпечення, створення надійних, складних паролів, а також використання багатофакторної аутентифікації для доступу. Окрім того, як організації, так і окремі користувачі мають постійно навчатися та підвищувати рівень своєї обізнаності в галузі кібербезпеки. Це потрібно для зменшення ймовірності використання зловмисниками людських помилок або технічних уразливостей з метою досягнення власних цілей [26].

1.3 Методи виявлення та запобігання загрозам

Все, що порушує роботу, цілісність або доступність IoT–пристрою чи мережі IoT–пристроїв, вважається загрозою. Загрози в середовищі IoT можуть мати різну природу та суттєво впливати на безпеку й стабільність системи. Їх можна розділити на три основні категорії: природні, ненавмисні та навмисні.

Природні небезпеки охоплюють катастрофічні події, наприклад, повені, землетруси, буревії або лісові пожежі. Ці небезпеки здатні фізично ушкодити

пристрої IoT, спричинити їх несправності та викликати тривалу непрацездатність мережевої інфраструктури. Хоча повністю уникнути таких загроз неможливо, їхній вплив можна зменшити за допомогою правильного резервного копіювання інформації, фізичного захисту приладів і планування відновлення після надзвичайних ситуацій [27].

Ненавмисні загрози виникають через випадкові помилки або недбалість користувачів і адміністраторів. Це можуть бути помилки під час налаштування мережі, випадкове видалення даних, недостатньо надійні паролі або некоректна конфігурація безпеки. Такий тип загроз часто недооцінюється, проте він може призвести до значних порушень роботи IoT–системи. Основними методами зниження ризику тут є навчання користувачів, регулярні перевірки конфігурації та впровадження автоматизованих механізмів моніторингу.

Навмисні загрози – найпідступніші, адже вони є результатом спланованих дій з боку злочинців. Мова йде про кібернетичні напади, метою яких є проникнення у пристрої, викрадення важливої інформації чи зрив функціонування мережі. До цього ж ряду зараховують DoS–атаки (відмова в обслуговуванні), інсталяцію шкідливих програм, а також спроби нелегального отримання доступу до IoT–обладнання. Щоб ефективно протистояти таким загрозам, критично важливо розгортати багатошарові системи безпеки, включаючи шифрування інформації, контроль прав доступу, систематичне оновлення програмних компонентів та застосування передових інструментів для виявлення загроз [28].

Ризики, пов’язані з IoT, поділяються на чотири основні категорії (рис. 1.2). До першої категорії належать ризики, пов’язані зі зломом пристроїв. Це включає втрату даних, фізичне пошкодження пристроїв та ризик порушення цілісності інформації, що обробляється. Наприклад, зловмисники можуть використовувати вразливості IoT–пристроїв для доступу до конфіденційних даних або навіть для впливу на фізичні процеси, керовані цими пристроями.

Інша категорія стосується спричинення збитків для інших зацікавлених сторін. Це може містити поширення шкідливого програмного забезпечення через пристрої IoT, використання їх як складових ботнетів для здійснення атак на інші системи, або випадкове порушення функціонування сторонніх інфраструктур. Цей ризик підкреслює необхідність впровадження механізмів, які пом'якшують вплив скомпрометованих пристроїв на інші вузли мережі.

Третя категорія охоплює ризик майбутньої експлуатації, коли зловмисники можуть використовувати IoT–пристрої як плацдарм для майбутніх атак. Це особливо актуально для пристроїв, які рідко оновлюються або мають обмежені ресурси для впровадження сучасних засобів захисту. Важливо, щоб розробники IoT–пристроїв враховували цей аспект ще на етапі проектування, забезпечуючи можливість оновлення та підтримки пристроїв протягом їхнього життєвого циклу.

Остання категорія стосується ризиків майбутньої агрегації, які виникають унаслідок взаємозалежності IoT-пристроїв і їхнього впливу на інші елементи мережевої інфраструктури. У великих масштабованих середовищах, де функціонують тисячі підключених пристроїв, навіть одна незначна вразливість може перетворитися на точку входу для широкомасштабної атаки. Наприклад, компрометація одного вузла може спричинити ланцюгову реакцію, що призведе до неконтрольованого поширення шкідливого трафіку, порушення цілісності даних або виведення з ладу критично важливих систем.

Подібні сценарії особливо небезпечні в умовах агрегації даних або централізованого керування, коли численні пристрої обмінюються інформацією або взаємодіють із загальними ресурсами. Тому виникає потреба в посилених превентивних заходах, спрямованих на стримування потенційного розповсюдження загроз.

Сегментація мережі, яка передбачає поділ інфраструктури на ізольовані логічні зони, дозволяє локалізувати загрозу та обмежити її вплив. Шифрування даних гарантує конфіденційність і захищає передану інформацію від перехоплення або модифікації. Контроль трафіку, зокрема за допомогою

фаєрволів, IDS/IPS та фільтрації за поведінковими ознаками, забезпечує виявлення аномальної активності ще на ранніх етапах її прояву [29].

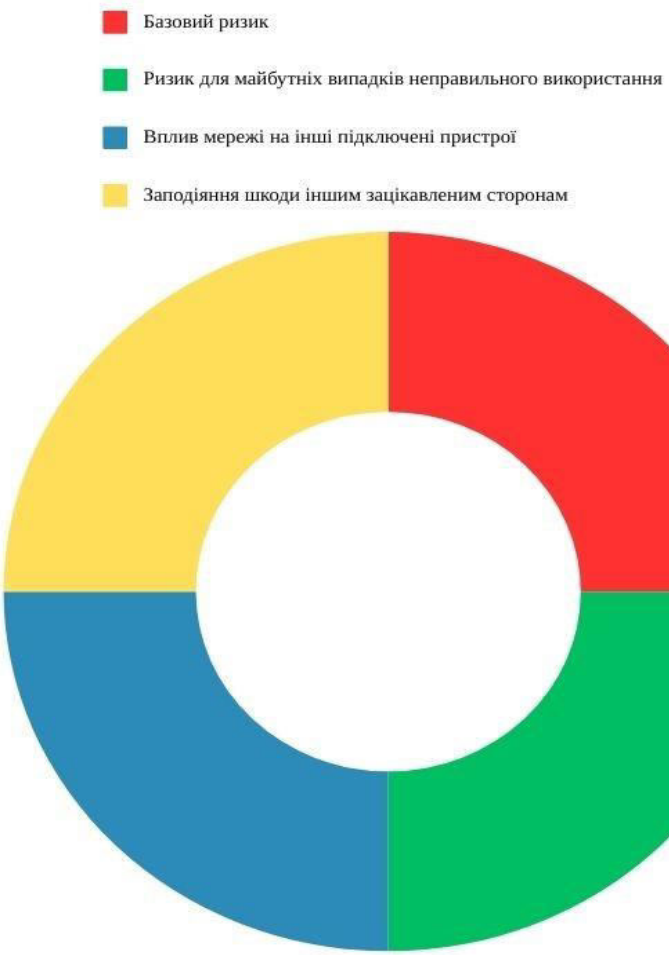


Рисунок 1.2 – Типи ризиків

Таким чином, загальні вимоги до безпеки систем IoT повинні включати наступне:

– доступність;

– аутентифікація;

– конфіденційність;

– стійкість.

Доступність – гарантія, що сервіси й ресурси (пристрої Інтернету речей) є доступними та зручними у використанні за запитом користувачів. Автентифікація – забезпечення, що тільки авторизовані особи мають змогу отримувати доступ до

пристроїв і послуг на основі авторизації (надання та скасування прав доступу), делегування (передання частини повноважень від одного об'єкта до іншого) і контролю автентичності (автентифікації) користувачів. Конфіденційність – захист існування з'єднань, потоків даних та інформаційного контенту від розголошення стороннім користувачам. Стійкість – забезпечення здатності даних та пристроїв IoT протистояти атакам і залишатися доступними навіть після атаки.

Головною метою систем виявлення та захисту від вторгнень (IDS) є забезпечення надійного захисту від ситуацій, які не відповідають нормальній роботі системи, але мають підозрілі характеристики, що можуть вказувати на неправомірне використання інформації або потенційні загрози. Такі системи розроблені для своєчасного виявлення атак, запобігання їм і підтримання стабільної безпеки навіть у разі виникнення подібних інцидентів.

Окрім виявлення й блокування загроз, системи виявлення вторгнень (IDS) відіграють ключову роль у фіксації відомих та нових типів нападів. Це сприяє безперервному поліпшенню механізмів оборони. Також, завдяки цьому, з'являється можливість аналізувати потенційні ризики та розробляти гнучкі стратегії безпеки, пристосовані до різноманітних ситуацій. IDS є важливим компонентом при розробці безпечних систем Інтернету речей, гарантуючи їхню здатність протистояти кібернетичним атакам [30].

Такі системи не тільки аналізують підозрілу активність, але й відстежують аномалії в мережевому трафіку, допомагаючи ідентифікувати потенційні загрози ще до їх реалізації. Вони можуть працювати як у реальному часі, так і в режимі ретроспективного аналізу, дозволяючи операторам досліджувати інциденти та розробляти відповідні заходи реагування. Крім цього, IDS інтегруються з іншими системами кібербезпеки, такими як файрволи, антивірусні програми та засоби шифрування даних, створюючи багаторівневу екосистему захисту. Це дозволяє не тільки оперативно реагувати на інциденти, але й мінімізувати їхній вплив на роботу організації чи мережі IoT.

Ключовим моментом функціонування систем виявлення вторгнень (IDS) є

їхня здатність розпізнавати раніше невідомі різновиди атак, використовуючи напрацювання машинного навчання та інтелектуальних систем. Це створює можливість проактивного захисту, коли система не просто реагує на загрози, а й передбачає їхній потенційний розвиток [31].

Системи виявлення вторгнень (IDS) надають інформацію про атаки, розширену діагностику, відновлення системи і ряд розслідувань, які дозволяють здійснювати поточні дії, такі як зупинка атак, припинення мережевих з'єднань або сеансів користувачів, блокування доступу до об'єкта атаки і виправлення відповідного програмного забезпечення. Коли IDS виявляє вторгнення в систему IoT, вона надсилає сповіщення у вигляді аудіо – або відеосигналу, електронного листа або текстового повідомлення на смартфон. Модифікацією цієї технології є система запобігання вторгненням (IPS), яка виявляє вторгнення і може запобігти їм за допомогою відповідних проактивних дій.

Компоненти системи виявлення та запобігання вторгненням (СВЗ) охоплюють декілька ключових складових, що забезпечують її функціонування на рисунку 1.3. Датчики або агенти – це модулі, що збирають інформацію про події та виконують початковий аналіз системи. У рамках СВЗ ці агенти називаються «сенсорами». Вони розміщуються в точках перехоплення трафіку для спостереження за мережевою активністю та виявлення потенційно загрозливих дій. Датчики можуть бути реалізовані як на апаратному, так і на програмному рівні, залежно від потреб системи.

Сервер управління виконує аналіз інформації, отриманої від датчиків, і визначає, чи відбулася атака. Для цього сервер використовує дані про поточну активність, а також інформацію з інших джерел, таких як підписи атак і профілі поведінки. Ця взаємодія дозволяє створити точний контекст для ідентифікації загроз. Сервер управління також взаємодіє з базою даних для збереження й подальшого аналізу отриманої інформації.

Інтерфейс управління, або консоль, забезпечує взаємодію між системою IPS і адміністратором. Консоль використовується для моніторингу подій, зібраних

системою, а також для конфігурації датчиків і оновлення програмного забезпечення. У деяких системах консоль також дозволяє здійснювати управління політиками безпеки, спрощуючи адміністрування [31].

База даних – це критично важливий компонент IPS, що відповідає за збереження відомостей, зібраних з сенсорів і керуючого сервера. Вона виступає як сховище підписів атак, моделей поведінки та журналів подій, також застосовується керуючим сервером для поглибленого аналізу даних. Централізована база даних збільшує продуктивність і точність системи, даючи змогу оперативно реагувати на загрози та складати звіти з безпеки.

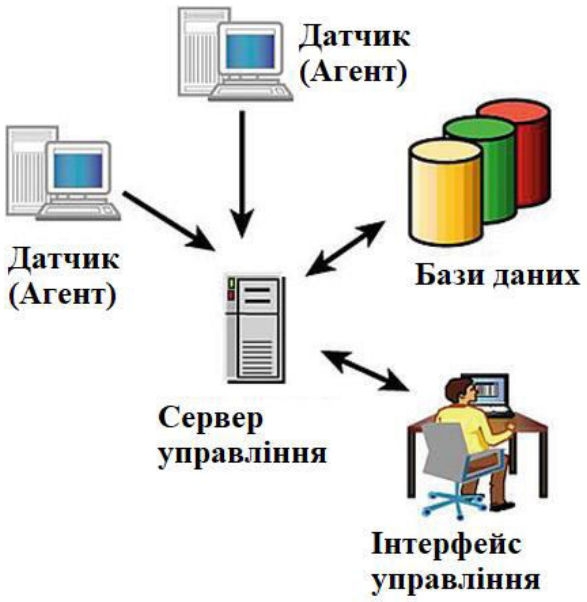


Рисунок 1.3 – Компоненти IPS

Існує два основних методи виявлення вторгнень:

– на основі сигнатур;

– на основі аномалій.

Сигнатурний спосіб детекції базується на зіставленні поточних подій з наперед встановленими сигнатурами, що відображають характерні риси вже відомих атак. Цей підхід характеризується високою ефективністю через простоту впровадження та відносно нескладний процес виявлення. Система аналізує

активність для відповідності збереженим зразкам, забезпечуючи точне розпізнавання раніше зафіксованих атак. Окрім того, метод формує розширені звіти про виявлені загрози, що робить його інформативнішим у порівнянні з моделлю, про яку йтиметься далі.

Однак ключовим недоліком такого підходу є його неспроможність виявляти нові або невідомі типи атак, адже система не володіє сигнатурами, які могли б описати їхню поведінку. Зокрема, сучасні види атак, такі як атаки «нульового дня» або складні багатовекторні загрози, залишаються поза сферою дії сигнатурних систем. Таким чином, сигнатурна модель є залежною від постійного оновлення бази сигнатур, що потребує значних зусиль з боку адміністраторів систем безпеки. Крім того, цей метод неефективний для виявлення загроз, які використовують раніше невідомі вразливості або обхідні техніки.

Метод виявлення на основі аномалій реалізується через аналіз поведінкових профілів, що репрезентують типову активність системи у штатному режимі. Профілі формуються для конкретних користувачів, мереж чи додатків шляхом відстеження їхньої діяльності протягом періоду, що називається навчальним або періодом оцінки. Ця модель проводить зіставлення поточної активності з раніше створеними профілями, виявляючи аномальну або нехарактерну поведінку.

Її головною перевагою є здатність виявляти раніше невідомі загрози, які не відповідають типовому профілю. Це робить аномалійний метод особливо корисним у середовищах із високим ризиком, де швидкість реагування на нові загрози критично важлива. Однак цей підхід має і свої недоліки. По–перше, велика кількість помилкових спрацьовувань може призводити до зайвого навантаження на команди безпеки. По–друге, створення точних профілів для складних або динамічних систем є складним завданням, яке потребує значних обчислювальних ресурсів. Нарешті, аномалійні системи можуть бути вразливими до атак, які поступово змінюють поведінку, адаптуючись до профілю (наприклад, атак типу «повільний перехід» або «мляве вторгнення») [32].

Однією з головних загроз для безпеки при розгортанні Інтернету речей (IoT) є використання заводських налаштувань пристроїв. Безпечне налаштування цих пристроїв передбачає налаштування шифрування для надійного зберігання та передачі інформації, впровадження безпечних протоколів з можливістю віддаленого доступу, а також забезпечення надійної автентифікації користувачів. Усі зміни в конфігурації необхідно ретельно задокументувати, а регулярні перевірки конфігурацій повинні підтверджувати, що безпечні параметри залишаються актуальними протягом усього терміну служби пристрою.

У випадку пристроїв IoT стандартні механізми IT–безпеки виявляються недостатніми, і ці пристрої вимагають спеціальних рішень у галузі мережевої безпеки. Одним із важливих підходів є сегментація мережі, яка дозволяє ізолювати IoT–пристрої від критично важливих бізнес–систем, тим самим знижуючи ризик потенційних порушень безпеки. Системи моніторингу мережевого трафіку допомагають виявляти аномальну поведінку пристроїв, що може свідчити про проблеми з безпекою. Незалежно від використовуваної платформи, команда безпеки повинна забезпечити використання зашифрованих VPN–з'єднань для безпечного віддаленого доступу, а також, за можливості, утримувати IoT–пристрої поза основною мережею [33].

Оновлення програмного забезпечення (прошивки) – це також ключовий елемент безпеки IoT. Систематичне оновлення дає змогу нейтралізувати відомі слабкі місця та мінімізувати ймовірність атак. Відтак, команди безпеки зобов'язані періодично перевіряти актуальність патчів і гарантувати їх своєчасне впровадження. Контроль доступу є критично важливим для захисту як фізичних, так і цифрових ресурсів. Використання надійних паролів та впровадження багатофакторної автентифікації допомагають гарантувати, що лише авторизовані користувачі мають доступ до пристроїв. Фізичні заходи контролю захищають IoT–пристрої від несанкціонованого втручання, а логічні обмеження запобігають небажаному віддаленому доступу. Безперервний моніторинг системи є ключовим елементом забезпечення безпеки IoT. Він дозволяє виявляти потенційні загрози

та оперативно на них реагувати. Такі системи мають відстежувати поведінку пристроїв, моделі мережевого трафіку та активність користувачів, щоб швидко ідентифікувати загрози. Регулярна оцінка рівня безпеки здатна виявляти вразливості ще до того, як ними скористаються зловмисники. Використання поведінкового аналізу для виявлення загроз дає можливість побудувати дієву систему захисту, що здатна пристосовуватися до нових типів атак. Автоматизовані способи аналізу сприяють зменшенню ролі людського фактора та прискорюють реакцію на інциденти. Впровадження подібних механізмів у систему моніторингу IoT–пристроїв, наприклад, термометрів у серверних кімнатах, не тільки збільшує рівень безпеки, але й забезпечує стабільну роботу критично важливої інфраструктури. Окрім цього, інтеграція механізмів машинного навчання та штучного інтелекту у процес моніторингу дозволяє суттєво підвищити точність виявлення аномалій і прогнозування потенційних загроз ще до їх реалізації. Завдяки здатності аналізувати великі обсяги телеметричних і поведінкових даних у реальному часі, такі алгоритми можуть виявляти нетипові відхилення, які важко вловити традиційними засобами.

Це не лише сприяє формуванню динамічних і гнучких політик безпеки, але й забезпечує їх автоматичне оновлення відповідно до змін у поведінці користувачів, пристроїв чи загального мережевого трафіку. Такі адаптивні механізми дозволяють мінімізувати кількість хибнопозитивних спрацювань, знижують навантаження на адміністратора та підвищують загальний рівень автономності системи.

Більше того, використання прогнозної аналітики на основі штучного інтелекту відкриває можливості для проактивного захисту – коли загроза нейтралізується ще до того, як вона завдасть шкоди. Таким чином, AI-компоненти не просто доповнюють існуючу інфраструктуру, а формують новий рівень інтелектуального кіберзахисту, здатного оперативно адаптуватися до швидко змінюваного середовища Інтернету речей [34].

2 РОЗРОБКА СИСТЕМИ ВИЯВЛЕННЯ ЗАГРОЗ НА ОСНОВІ ПОВЕДІНКОВОГО АНАЛІЗУ

2.1 Архітектра системи виявлення загроз для IoT–пристроїв

Архітектура системи зображена на рисунку 2.1, призначеної для виявлення загроз у світі IoT, мусить зважати на обмежені обчислювальні можливості пристроїв, різноманітність мережевих оточень та потребу в оперативному аналізі трафіку. Головними критеріями для оцінки такої системи є висока точність ідентифікації атак, здатність адаптуватися до нових небезпек, а також мінімальний вплив на функціонування інфраструктури IoT. В даному розділі будуть розглянуті основні підходи до формування архітектури систем виявлення загроз для IoT–пристроїв, її складові частини та способи забезпечення безпеки.

IoT–пристрої, зокрема термометри, що використовуються для моніторингу температури в серверних приміщеннях, є потенційними векторами атак у межах інформаційної системи. Через обмежені обчислювальні ресурси та спрощені механізми автентифікації ці пристрої часто мають низький рівень захисту, що робить їх уразливими до різноманітних загроз.

Одним з найважливіших викликів є скомпрометованість автентифікації, що здатна привести до нелегального доступу до вашого приладу. Застосування звичайних або надто простих паролів, невикористання двофакторної перевірки, та неефективне регулювання доступом створюють умови, в котрих зловмисник може отримати керування термометром. Це можливо через атаку методом перебирання паролів, або через використання слабких місць в механізмах авторизації [35]. Ще однією загрозою є можливість перехоплення та модифікації трафіку, що виникає при використанні незашифрованого або слабко зашифрованого каналу зв’язку. У такому випадку зловмисник, здійснюючи атаку типу «людина посередині» (Man–in–the–Middle), може отримати доступ до переданих даних, підміняти їх або перехоплювати управлінські команди. Це може

призвести до викривлення інформації про температуру та створення хибних тривог або, навпаки, приховування критичних змін у серверному приміщенні.

Вразливості, які можуть бути у програмному забезпеченні IoT–термометра, становлять потенційну загрозу. Якщо пристрій використовує стару версію програмного забезпечення або має невідомі дірки, зловмисник може спробувати використати їх, щоб запустити свій код дистанційно. Це може призвести до можливості розширити атаку на інші пристрої у мережі. Ще однією поширеною загрозою є атаки на мережевий стек, такі як відмова в обслуговуванні (Denial of Service, DoS). Генеруючи велику кількість запитів до пристрою, зловмисник може спричинити його перевантаження, що унеможливить коректну передачу даних та ускладнить адміністрування [36].

Також можлива фальсифікація даних, коли зловмисник, отримавши контроль над пристроєм або каналом передачі, змінює показники температури. Це може впливати на автоматизовані системи клімат–контролю, створюючи умови, за яких серверне обладнання буде працювати в неконтрольованому середовищі, що, у свою чергу, може призвести до його перегріву та виходу з ладу.

Крім того, IoT–термометр здатний виступати «дверима» для бокових атак на ключову мережу установи. Якщо прилад не має чітко обмеженого доступу до інших складових системи, зловмисник має змогу використати його для подальшого проникнення до серверної інфраструктури, зчитування важливих даних або розповсюдження шкідливого ПЗ. Таким чином, серверна частина системи виконує ключову роль у забезпеченні комплексної безпеки IoT–інфраструктури. Її функціональність виходить далеко за межі простого накопичення інформації. Завдяки здатності обробляти великі обсяги даних у реальному часі, сервер виступає як аналітичний та координаційний центр, який поєднує в собі інструменти для виявлення аномалій, адаптації політик безпеки, а також динамічного реагування на загрози. Його здатність накопичувати історичні дані про поведінку пристроїв дає змогу виявляти не лише явні атаки, а й повільні, малопомітні зміни, які можуть бути попередниками більш серйозних інцидентів.

Важливим аспектом є те, що сервер забезпечує не просто реактивну, а проактивну модель захисту. Це означає, що на основі накопиченої інформації, аналітики та змін у поведінці пристроїв, система може самостійно оновлювати політики безпеки, оптимізувати порогові значення параметрів, виявляти потенційно небезпечні тенденції. Такий підхід дозволяє не лише миттєво реагувати на загрози, а й попереджати їх ще до початку активної фази атаки [37].

Ще однією важливою перевагою бекенд–частини системи є її здатність миттєво реагувати на нові виклики та зміни в середовищі функціонування. У стрімкому світі IoT, де пристрої постійно додаються, переміщуються або змінюють сценарії свого використання, критично важливо, щоб система демонструвала високу адаптивність і могла оперативно оновлювати свої механізми без потреби в ручному втручанні з боку адміністратора.

Централізоване зберігання моделей поведінки дозволяє серверу підтримувати єдину базу норм і аномалій, забезпечуючи їх актуалізацію в реальному часі. Завдяки можливості гнучкого та автоматизованого оновлення моделей, серверна частина виконує роль постійного координатора безпеки, здатного забезпечити стабільний захист навіть за умов високої динаміки мережевого середовища або при масштабуванні інфраструктури.

Крім того, надзвичайно важливою функцією є повноцінне журналювання всіх подій, пов’язаних із поведінкою пристроїв та реакціями системи. Серверна частина реєструє кожну дію, кожне відхилення від очікуваних параметрів, а також усі спроби зовнішнього або внутрішнього втручання в роботу пристроїв. Такий підхід забезпечує не лише ретельне постінцидентне розслідування, а й створює умови для довгострокового удосконалення системи безпеки.

Аналіз накопичених логів дає змогу виявляти приховані або раніше невідомі патерни аномальної активності. Це дозволяє розширювати базу знань системи, вчасно виявляти нові вектори атак та покращувати ефективність механізмів виявлення загроз. Таким чином, бекенд виступає не лише як центр

управління та зберігання, а й як аналітичний вузол, що постійно навчається та адаптується, підвищуючи загальний рівень захисту мережі.

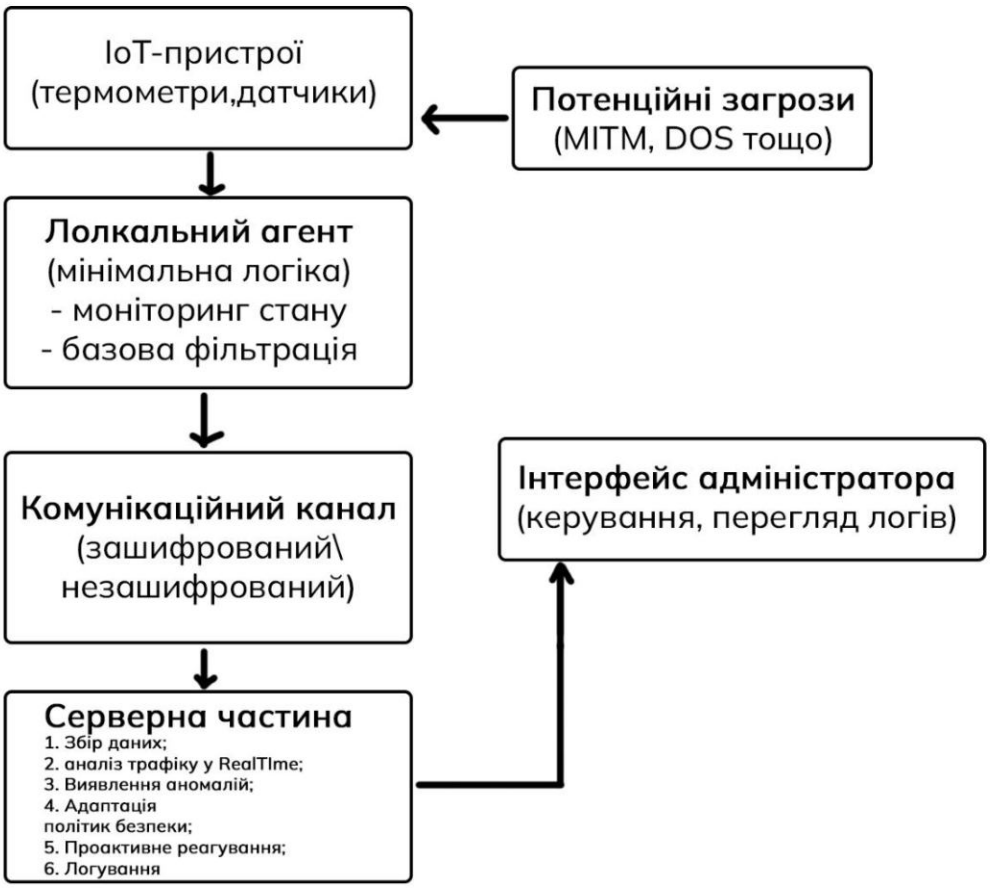


Рисунок 2.1 – Архітектура системи виявлення загроз

Принцип роботи системи виявлення загроз для IoT–пристроїв ґрунтується на багаторівневій обробці даних і аналізі поведінки пристроїв у мережі. На першому рівні IoT–пристрої (наприклад, термометри або датчики) виконують безперервний збір даних про параметри середовища. Ці дані надходять до локального агента – модуля з мінімальною логікою, який розташований безпосередньо біля пристрою. Завдання локального агента полягає у базовому моніторингу стану пристрою та попередній фільтрації даних, щоб зменшити обсяг зайвої або шкідливої інформації, яка передається далі.

Після цього інформація передається через комунікаційний канал. Передача може відбуватися як у зашифрованому вигляді, так і у відкритому, залежно від

рівня безпеки, що реалізується у конкретному випадку. Комунікаційний канал є потенційною точкою атаки, зокрема для атак типу MITM або DOS.

Дані надходять до серверної частини системи, яка є її аналітичним ядром. Сервер виконує кілька ключових функцій: збір даних від усіх пристроїв, аналіз трафіку в реальному часі, виявлення аномальної поведінки, діагностику можливих загроз і відхилень від очікуваної поведінки пристроїв. На основі цих процесів формуються політики безпеки, здійснюється реагування на інциденти, а також ведеться логування для подальшого аналізу.

Крім того, система має інтерфейс адміністратора, за допомогою якого здійснюється керування системою, перегляд логів та прийняття рішень щодо відповіді на загрози. Увесь процес спрямований на виявлення шкідливої активності в поведінці пристроїв і забезпечення стабільної та безпечної роботи IoT–інфраструктури.

2.2 Контролер IoT–мережі, функції системи виявлення загроз

Контролер, як ключовий елемент у системі виявлення загроз, виступає локальним осередком для обробки інформації та ухвалення рішень. Головна його функція полягає у безперервному відстеженні активності підключених пристроїв IoT, швидкому виявленні аномалій на рівні окремих елементів мережі та мінімізації ймовірних ризиків ще до того, як загроза зможе розповсюдитись по інфраструктурі. Завдяки безпосередній близькості до фізичних пристроїв, контролер має можливість оперувати «сирими» даними в реальному часі, блискавично реагуючи навіть на незначні зміни у звичній поведінці.

Функціональність контролера охоплює декілька рівнів. На першому рівні він виступає як колектор даних – зчитує пакети трафіку, параметри сенсорів, команди управління, часові мітки, рівень споживання ресурсів (наприклад, енергії чи процесорного часу), а також інші ключові індикатори активності пристроїв.

Зібрані дані фіксуються у тимчасовому буфері або одразу передаються до модуля аналізу поведінки. Важливо, що саме на цьому етапі контролер може здійснювати попередню фільтрацію, видаляючи «шумові» сигнали, або навпаки – виділяючи підозрілі шаблони для глибшого аналізу.

Контролер, як ключовий вузол в системі детектування загроз, є локальним центром обробки даних і прийняття рішень. Його основна роль – постійний моніторинг активності приєднаних IoT–пристроїв, швидке виявлення аномалій на рівні окремих компонентів мережі та зменшення потенційних ризиків ще до поширення загрози по всій інфраструктурі. Завдяки безпосередньому розташуванню поблизу фізичних пристроїв, контролер може працювати з «сирими» даними в реальному часі, миттєво реагуючи на найменші зміни у звичній роботі.

Окрему роль відіграє можливість контролера самостійно втручатися у роботу підключених пристроїв. У разі підтвердження загрози він може призупинити активність підозрілого пристрою, відключити його від мережі або змінити маршрут передавання даних для мінімізації потенційної шкоди. Такий підхід забезпечує не лише пасивний, а й активний захист – тобто здатність протидіяти атакам без очікування інструкцій від центрального сервера [38].

Контролер додатково відповідає за ведення локальних журналів подій. Усі здійснені дії, інциденти, відхилення від норми та отримані сигнали реєструються з метою їх подальшого аналізу або синхронізації з серверною складовою системи. У випадку втрати зв’язку з сервером, контролер може функціонувати автономно, забезпечуючи тимчасовий захист мережі до моменту відновлення з’єднання [39].

Важливо, що контролер не лише працює з фіксованими правилами, а й здатен адаптуватися до змін у поведінці пристроїв, оновлюючи моделі «нормальної» поведінки. Це особливо критично в умовах, коли IoT–пристрої працюють у змінному середовищі, де характер трафіку та інтенсивність роботи можуть варіюватися залежно від часу доби, навантаження або зовнішніх

чинників. Така адаптивність значно знижує ймовірність помилкових спрацювань і дозволяє краще виявляти нові, ще невідомі типи атак.

Крім того, керуючий вузол здійснює взаємодію з суміжними блоками – скажімо, системами ідентифікації, шифрування передачі даних, засобами ведення логів, системами клімат–контролю, а також іншими контролерами в межах локальної мережі на рисунку 2.2. Така розподілена взаємодія дає можливість виявляти атаки, які характеризуються складною багатофазною структурою, коли кожен окремий крок виглядає нешкідливим, проте в сукупності формує загрозу.

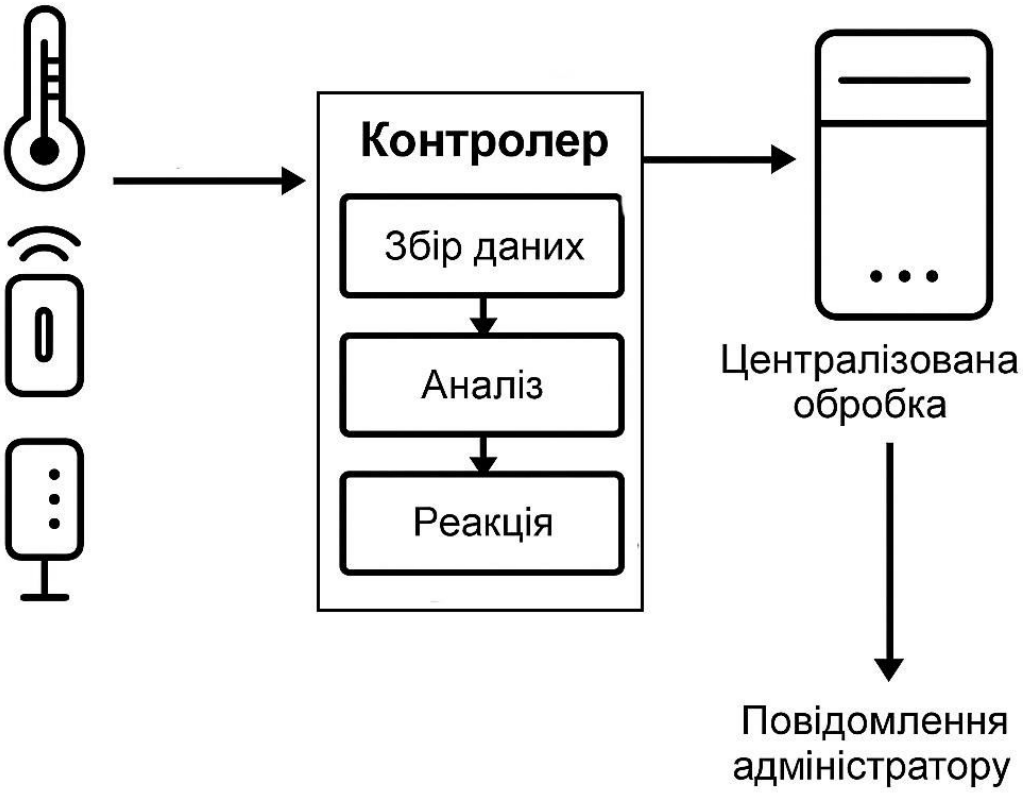


Рисунок 2.2 – Функціональна схема роботи контролера системи виявлення загроз у IoT – мережі

На схемі зображено спрощений принцип роботи системи виявлення загроз для IoT–пристроїв. Дані з термометрів або сенсорів надходять до контролера, який виконує три основні функції: збір інформації, її аналіз та реагування на виявлені аномалії. Далі дані передаються до централізованої обробки на сервері,

де проводиться додатковий аналіз, після чого адміністратор отримує відповідні повідомлення про події чи загрози.

2.3 Визначення нормальної поведінки пристроїв в IoT–мережі

Визначення нормальної поведінки пристроїв в IoT–мережі є ключовим етапом у створенні ефективної системи виявлення загроз на основі поведінкового аналізу. Оскільки IoT–пристрої виконують чітко визначені функції в стабільному середовищі, їх поведінка зазвичай має регулярний, передбачуваний характер. Це дозволяє сформувати базовий профіль «норми», з яким можна порівнювати поточну діяльність пристроїв для виявлення аномалій.

Звичайна поведінка встановлюється шляхом аналізу параметрів, що описують функціонування пристрою. Скажімо, для IoT–термометра це можуть бути такі показники: частота відправки даних, середнє значення температури, стійкість сигналу, тривалість з'єднання, час відгуку пристрою, а також характеристики трафіку (об'єм, протокол, напрямок передачі). Для кожного параметра визначається статистичний діапазон дозволених значень, розрахований на основі зібраних даних за певний період без ознак атак.

Процес побудови профілю нормальної поведінки включає декілька етапів:

– збір даних протягом навчального періоду;

– їх попередню обробку та фільтрацію;

– побудову моделей на основі середніх значень, стандартних відхилень та часових закономірностей;

– визначення граничних значень, за межами яких поведінка вважається підозрілою.

Наприклад, якщо термометр передає дані кожні 30 секунд з допустимим діапазоном температур від 15 до 30°C, відхилення понад 10% за частотою

передачі або раптове зростання температури вище 35°C можуть бути розцінені як аномалії.

Система повинна враховувати також контекст роботи пристрою. Наприклад, коливання температури вдень і вночі є нормальними, тож для зменшення кількості хибних спрацювань потрібно будувати модель, яка враховує час доби та сезонні фактори. Крім того, важливо адаптувати модель до змін у мережі – наприклад, після оновлення прошивки або зміни конфігурації пристрою.

Варто зосередитись на взаємовідносинах між пристроями – якщо один пристрій починає надсилати дані іншим, чого раніше не було, це може свідчити про можливе порушення безпеки. Отже, «норма» формується не лише на рівні окремих пристроїв, а й з урахуванням загальної топології та особливостей взаємодії в IoT–мережі. Завдяки поведінковому аналізу можна виявляти не лише відомі шаблони атак, а й нові, раніше невідомі загрози. Побудова точного профілю нормальної поведінки є фундаментом для цього підходу, оскільки вона дозволяє системі виявляти навіть найменші відхилення від звичного функціонування, що може свідчити про потенційну загрозу, і саме ця особливість дає йому перевагу над класичними методами сигнатурного виявлення загроз, які здатні реагувати лише на вже відомі атаки.

2.4 Визначення допустимих відхилень у поведінці пристроїв

У контексті поведінкового аналізу в IoT–системах критично важливо визначити допустимі межі для нормальної роботи пристроїв. Чітке окреслення таких меж дозволяє уникнути як хибнопозитивних, так і хибнонегативних спрацювань системи виявлення загроз, що, у свою чергу, забезпечує стабільність функціонування всієї інфраструктури. Допустимі межі встановлюються на основі історичних даних, контексту використання та типових сценаріїв поведінки пристрою, що дозволяє системі гнучко реагувати на природні варіації без

надмірної чутливості до незначних відхилень. Крім того, ці межі повинні адаптуватися в динаміці, враховуючи оновлення програмного забезпечення, зміну режиму роботи чи зміну ролі пристрою в мережі, що забезпечує довготривалу актуальність поведінкової моделі. Таким чином, грамотне визначення меж «норми» є основою для точного, адаптивного та ефективного виявлення загроз у середовищі IoT. Для IoT–термометра, як приклад пристрою з обмеженим функціоналом, до моніторингу можуть бути включені такі параметри:

– температура навколишнього середовища (T) – вимірюється кожні n секунд;

– частота надсилання даних (f) – кількість запитів до сервера на годину;

– розмір переданого пакету (S) – у байтах;

– час відповіді пристрою (RTT) – у мілісекундах;

– використання енергоспоживання (P) – для автономних IoT–девайсів.

Для кожного параметра визначається базове значення X_{avg} – середнє значення за певний період, та граничне відхилення ΔX, що допускається без генерації тривоги:

$$X_{min} = X_{avg} - \Delta X \quad (2.1)$$

$$X_{max} = X_{avg} - \Delta X \quad (2.2)$$

Де, X_{avg} – середнє значення за певний період, та граничне відхилення ΔX, що допускається без генерації тривоги.

Відповідно, можна навести приклад для температури:

$$T_{avg} = 24°C, \quad \Delta T = 2°C \Rightarrow [T_{min}, T_{max}] = [22°C, 26°C] \quad (2.3)$$

Де, T_{avg} – середнє значення, ΔT - граничне відхилення.

Якщо $T_{current} > T_{max}$ або $T_{current} < T_{min}$, система фіксує аномалію.

Відповідно класифікація відхилень за рівнем критичності виглядатиме наступним чином:

– мінімальні (допустимі) – до ±5% від середнього значення;

– середні (помірні) – від ±5% до ±10%, фіксуються в логах;

– критичні – понад ±10%, ініціюється реакція системи.

Отже, можна створити таблицю допустимих відхилень:

Таблиця 2.1 – Допустипі відхилення у поведінці пристроя

Параметр	Середнє значення	Допустиме відхилення (±%)	Межі
Температура (T)	24°C	±10% (≈2.4°C)	[21.6°C, 26.4°C]
Частота запитів (f)	60/год	±10% (6 запитів)	[54, 66]
Розмір пакету (S)	512 байт	±5% (≈26 байт)	[486, 538]
Час відповіді (RTT)	100 мс	±20% (20 мс)	[80 мс, 120 мс]

Щоб не реагувати на одноразові сплески, система аналізує ковзаюче вікно останніх значень:

$$X_{avg}(t) = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{2.3}$$

де $X_{avg}(t)$ – середнє значення ознаки в момент часу t, обчислене за останні nnn вимірювань (усереднене значення в ковзаючому вікні), n – кількість останніх спостережень, X_i – значення ознаки на і–му кроці всередині вікна спостереження. $\sum_{i=1}^{n} X_i$ – сума n останніх значень ознаки.

Це дозволяє фільтрувати випадкові флуктуації, реагуючи тільки на системні зміни на рисунку 2.4.

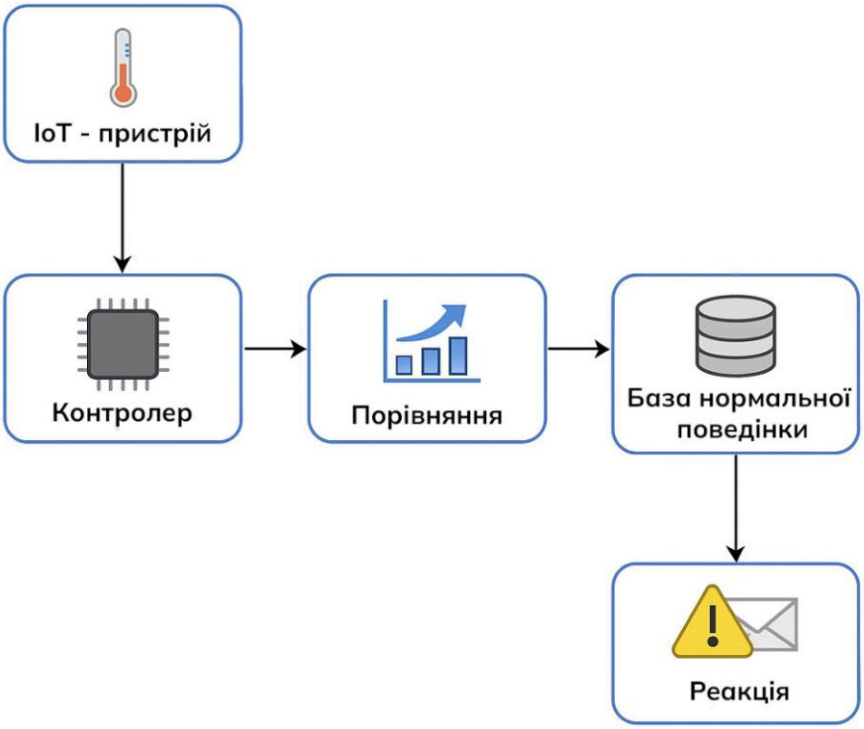


Рисунок 2.4 – Схема роботи виявлення аномалій

Ця схема ілюструє процес обробки даних з IoT–пристрою. Спочатку пристрій збирає інформацію, наприклад, про температуру навколишнього середовища. Потім ці дані передаються до контролера, який виконує первинну обробку. Далі отримані значення порівнюються з еталонними, що зберігаються у базі нормальної поведінки. Якщо система виявляє відхилення від норми, вона активує відповідну реакцію – наприклад, надсилає сповіщення або сигнал тривоги. Такий підхід дозволяє виявляти аномалії в роботі пристрою та швидко реагувати на потенційні проблеми.

2.5 Реакція на аномалії та оновлення політик безпеки

Реагування на виявлені аномалії на рисунку 2.5 – це критично важлива складова системи аналізу поведінки, яка гарантує не лише ідентифікацію загроз, а й оперативну та дієву відповідь на них. Система, що тільки реєструє відхилення, але не має механізмів впливу на мережу, не здатна забезпечити захист у

швидкоплинному середовищі IoT. Відтак, надзвичайно важливим є впровадження повного циклу: виявлення – реагування – адаптація.

Коли система виявляє відхилення від нормальної поведінки пристрою, контролер або серверна частина ініціює аналіз серйозності інциденту. Залежно від характеру аномалії, її масштабу, частоти та контексту виникнення, обирається рівень реагування. Невеликі відхилення можуть бути зареєстровані лише в журналі подій, тоді як значні або повторювані – вимагають активних дій. Важливо, що система має бути не лише чутливою, а й розумною, тобто такою, що вміє диференціювати випадкові флуктуації від реальних загроз.

Реагування може втілюватися у різних способах. Найбільш простий – це сповіщення адміністратора про виявлену нештатну ситуацію. Таке сповіщення здатне включати в себе: ідентифікатор пристрою, показники, які стали причиною спрацювання, поточне значення, граничне значення, ступінь ризику та поради щодо подальших кроків. Якщо ж система має налаштування для автономної реакції, вона може застосувати заходи локального чи глобального масштабу: ізолювати пристрій від мережі, обмежити його функціонал, вимкнути або перевести у безпечний режим [40].

На серверному рівні система може оновлювати політики безпеки, які зберігаються у централізованій базі, це оновлення може мати кілька форм.

З метою підвищення гнучкості та точності системи виявлення загроз передбачено кілька механізмів адаптації моделі поведінки. По–перше, можливе динамічне коригування меж допустимих значень: якщо після оновлення пристрій починає працювати в новому режимі, система здатна автоматично підлаштувати модель під змінені умови. По–друге, до профілю поведінки можуть додаватися нові ознаки у разі виявлення нової, сталої та безпечної активності. Нарешті, здійснюється регулярне оновлення шаблонів аномалій – до бази вносяться нові поведінкові патерни, що потребують моніторингу в майбутньому.

Цей процес можна назвати адаптивним захистом, коли система не лише виявляє і реагує, а й навчається на основі отриманого досвіду. У цьому полягає

основна перевага поведінкового аналізу над традиційними сигнатурними системами: замість фіксованого списку загроз система має динамічну модель поведінки, яка може змінюватися в режимі реального часу.

Окрім зазначеного, ведення журналів подій є не менш ключовим. Усі відхилення від норми, навіть якщо вони не потребували миттєвої реакції, фіксуються у вигляді логів. Це дає можливість згодом проводити ретроспективний аналіз, знаходити довгострокові тенденції, покращувати фільтри виявлення або ж формулювати нові припущення стосовно поведінки пристроїв.

Політики безпеки мають бути єдиними та узгодженими у межах всієї IoT–інфраструктури. З цією метою сервер після оновлення конфігурації розповсюджує її до всіх контролерів, які працюють у мережі. Такий централізований підхід дозволяє уникнути конфліктів політик, несумісності фільтрів або нерівномірного рівня захисту.

Реагування системи також має зважати на контекст, в якому відбувається подія. Скажімо, однакові зміни у показниках можуть мати абсолютно різне значення в залежності від часу доби, типу дня (робочий чи вихідний), сезону, а також поточного режиму роботи системи – наприклад, у періоди пікового навантаження чи під час планового технічного обслуговування. Без урахування контексту система може некоректно інтерпретувати поведінку як аномальну, що призведе до хибного спрацювання або, навпаки, до ігнорування дійсної загрози.

Тому доцільно впроваджувати контекстно–орієнтовані правила реагування, які враховують зовнішні обставини, календарні фактори, профіль користувача, тип пристрою, характер його взаємодії з іншими вузлами та навіть геолокаційні умови. Такі правила можуть ґрунтуватися на аналізі попередньої історії подій, шаблонах типових сценаріїв та статистичних залежностях, що дозволяє системі приймати зважені рішення на основі більш повної картини.

Контекстно-залежне реагування підвищує точність роботи механізмів захисту, мінімізує кількість помилкових втручань у нормальну діяльність

пристроїв і забезпечує більшу довіру з боку користувачів та операторів. У кінцевому підсумку це сприяє створенню більш стабільного, розумного та саморегульованого середовища IoT.

Рисунок 2.5 – Схема реації виявлення аномалій та оновлення політик безпеки

Ця схема описує процес реагування на аномалії в системі безпеки. Спочатку система виявляє аномалію – відхилення від нормальної поведінки. Далі відбувається аналіз інциденту для з’ясування причин і потенційних загроз. Після цього приймається рішення: чи потрібно реагувати на цю аномалію. Якщо ні – цикл починається знову з етапу виявлення. Якщо так – реалізується відповідна реакція (наприклад, блокування доступу або повідомлення адміністратора). У разі підтвердженої загрози оновлюється політика безпеки, щоб у майбутньому уникнути подібних ситуацій.

У другому розділі представлено концептуальне бачення створення системи детекції загроз у мережах IoT на основі аналізу поведінкових патернів. Проведене дослідження дало змогу окреслити архітектурні характеристики, визначити функції основних компонентів системи та принципи їхньої взаємодії. Це критично важливо для своєчасного виявлення й оперативного реагування на потенційні загрози.

Під час розробки архітектури враховано обмежені обчислювальні ресурси IoT–пристроїв, специфіку їхньої роботи в неоднорідному середовищі та потребу у швидкому відгуку. Доведено, що безпека має реалізовуватися не лише на серверному рівні, а й у вигляді розподіленої моделі, де контролер виконує активну роль – фільтрує, виявляє та блокує нетипову активність. Контролер виступає ключовим елементом реагування: відстежує телеметрію, реєструє відхилення, зберігає журнали, передає дані на сервер і може самостійно реагувати на загрози. Він також координує взаємодію з іншими компонентами – засобами шифрування, аутентифікації та адміністрування.

Окрему увагу приділено виявленню типової поведінки IoT–пристроїв, що є основою поведінкового аналізу. Запропоновано метод побудови профілю «норми» на базі статистичних моделей, які враховують ковзне середнє, допустимі відхилення та контекстні фактори. Система класифікує аномалії за рівнем загрози, забезпечуючи гнучке реагування та зменшуючи кількість хибних спрацьовувань.

Також розглянуто механізми реагування на інциденти: блокування пристроїв, інформування адміністратора, корекція політик безпеки та оновлення профілів поведінки. Представлено концепцію адаптивного захисту, за якої система вдосконалюється на основі накопиченого досвіду – формує нові правила та оптимізує алгоритми виявлення.

3 РЕАЛІЗАЦІЯ ПРОТОТИПУ СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ

3.1 Вибір середовища розробки та інструментів

Для реалізації системи виявлення загроз у IoT–мережі було обрано прості, доступні та добре документовані інструменти, які дозволяють створити діючу модель без складних налаштувань і додаткового навчання.

Як контролер, застосовано Arduino Nano або ESP8266, адже ці плати доступні за ціною, заощаджують енергію та просто програмуються. Для збирання даних застосовано простий цифровий термометр (скажімо, DS18B20), який передає показники температури кожні пів хвилини. Програмування контролера виконується в Arduino IDE, що надає підтримку усіх необхідних бібліотек, має зрозумілий синтаксис та дозволяє швидко налаштовувати пристрої через USB–з'єднання.

Контролерна частина була реалізована на основі плати ESP8266 та цифрового термодатчика DS18B20. Зчитані дані обробляються у мікроконтролері, де формується буфер з останніх десяти значень температури, обчислюється середнє значення, і в залежності від відхилення, формується статус – «ok» або «anomaly» (рис. 3.2, 3.3). Передача даних відбувається через Wi–Fi, у вигляді HTTP POST–запитів з JSON–структурою.

Функціонал реалізовано за допомогою платформи Arduino IDE та стандартних бібліотек таких як:

– ESP8266WiFi.h;

– OneWire.h;

– DallasTemperature.h;

– ESP8266HTTPClient.h.

Для з'єднання датчика температури DS18B20 з мікроконтролером ESP8266 (NodeMCU) знадобиться три проводи: живлення (VDD), загальний провід (GND) та лінія передачі даних (DATA). Живлення датчика (VDD) під'єднується до виходу 3.3V на платі ESP8266, контакт GND – до «землі» (G), а контакт DATA –

до цифрового входу D4 (GPIO2). Для коректної роботи необхідно обов'язково встановити підтягувальний резистор на 4.7 кОм між лінією DATA та живленням (VDD). Завдяки такій схемі (рис. 3.1) можливо стабільно отримувати показники температури з датчика передавати контролеру для подальшої обробки інформації.

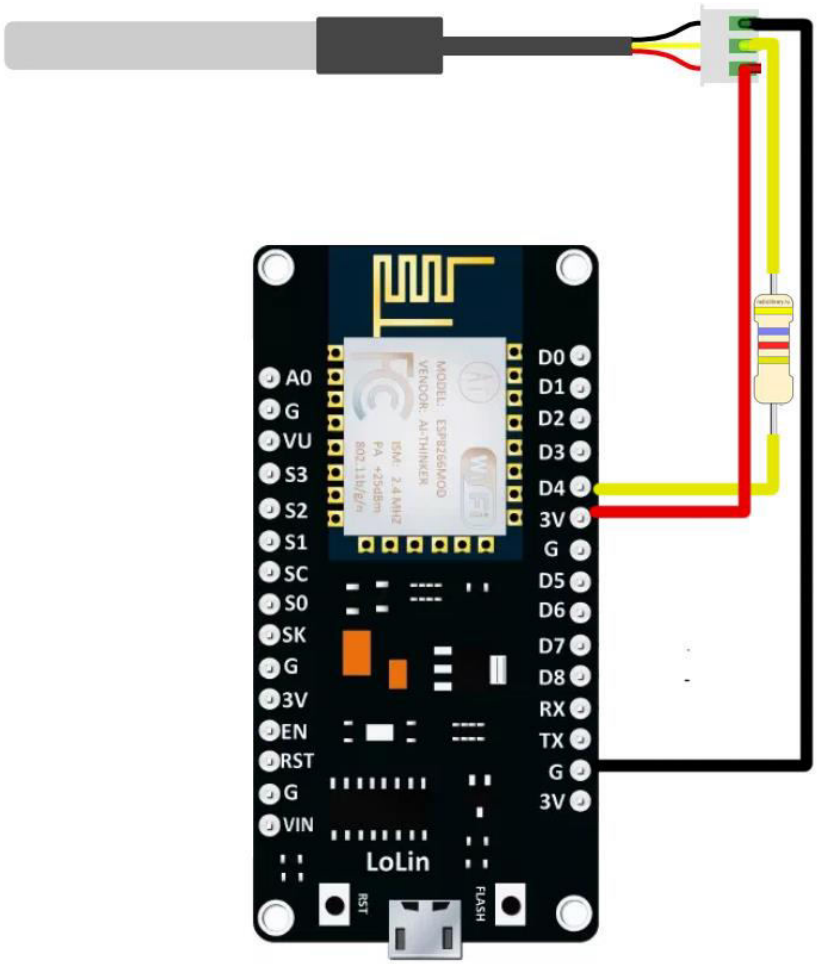


Рисунок 3.1 – Схема підключення датчика

Для з'єднання з мережею та відправлення інформації на сервер використовується бібліотека ESP8266WiFi, а для передачі даних – протокол HTTP POST (або MQTT, якщо буде розширення проекту). Інформація передається у форматі звичайного тексту або як JSON–структура.

Аналіз відхилень базується на простій логіці: якщо температура виходить за дозволені рамки (наприклад, ±10% від середнього показника), сервер визначає аномалію та видає попередження у консолі або фіксує запис у логах. Для

вирішення поставленого завдання, були застосовані такі інструменти та бібліотеки:

– Arduino IDE – середовище для написання прошивки для контролера;

– ESP8266WiFi / WiFi.h – підключення до Wi–Fi;

– Flask (Python) – мінімалістичний HTTP–сервер;

– json / time (Python) – для обробки даних на сервері;

– Visual Studio Code / Thonny – середовище для редагування коду на Python.

При такому підході система є легкою в реалізації, не вимагає потужного обладнання та може бути розгорнута навіть на домашньому рівні для демонстрації принципів поведінкового аналізу та базового реагування на аномальні події.

3.2 Архітектура програмного забезпечення

Програмна архітектура системи виявлення загроз на основі поведінкового аналізу побудована з урахуванням максимальної простоти реалізації, мінімального використання складних бібліотек і максимальної наочності процесів. Основна мета цієї реалізації – показати принцип дії системи у реальних умовах, використовуючи доступні апаратні засоби та базові програмні інструменти.

Система складається з двох головних частин:

– контролерна частина;

– серверна частина.

Контролерна частина встановлюється безпосередньо на IoT–пристрій (ESP8266, Arduino), який виконує зчитування температури та початкову обробку даних. Серверна частина – працює на окремому пристрої де виконується аналіз, збереження та реакція на аномальні дані. Ці два компоненти спілкуються один з одним через Wi–Fi, надсилаючи звичайні HTTP–запити. Обмін даними

реалізується за допомогою JSON–структури, що являє собою сучасний та практичний формат, з яким без проблем працюють у більшості мов програмування.

Контролерна секція збудована на мікроконтролері ESP8266, з'єднаному з цифровим датчиком температури DS18B20. Для того, аби не тільки передавати дані, але й здійснювати елементи аналізу поведінки, контролер зберігає послідовність останніх значень температури в буфері – скажімо, 10 останніх показів.

На кожному циклі контролер:

– зчитує температуру;

– додає значення в буфер;

– розраховує середнє значення;

– порівнює нові дані з середнім значенням ±10%;

– якщо є відхилення – відзначає його як аномалію (навіть без сервера);

– формує JSON–об'єкт із поточною температурою та статусом («ok» або «anomaly»);

– надсилає дані HTTP POST–запитом на локальний сервер.

Це знижує навантаження на сервер, адже вже на рівні контролера можна відсіяти нормальні дані або провести первинну фільтрацію.

Сервер працює на Python з використанням бібліотеки Flask, яка дозволяє створювати простий веб–сервер у кількох рядках коду. Він слухає HTTP–запити на певному порту (наприклад, 5000), приймає JSON–дані від контролера, розбирає їх і виконує перевірку

Серверна логіка включає:

– прийом температурного значення та статусу;

– додаткову перевірку на аномалію;

– порівняння з допустимими межами (наприклад, 20–26°C);

– логування подій – як нормальних, так і підозрілих;

– вивід повідомлення у консоль або на веб–інтерфейс (опційно).

Обмін даними між компонентами реалізується через HTTP POST запити. Контролер, підключений до мережі Wi–Fi, регулярно (скажімо, кожні півхвилини) відправляє запит на локальну IP–адресу серверу.

На рисунку 3.1 та 3.2 зображений типовий JSON–пакет, що надсилається.

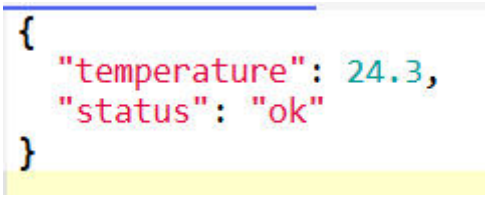


Рисунок 3.2 – Json пакет який вказує на нормальність

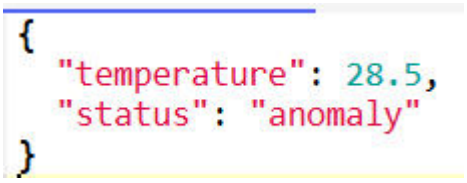


Рисунок 3.3 – json пакет який вказує на аномальість

Сервер одразу розуміє, що сталося, і може швидко відреагувати. Якщо потрібно, до JSON можна додати час, ID пристрою або інші параметри.

3.3 Реалізація контролерної частини

Контролерна частина системи є першою лінією моніторингу та виявлення потенційних відхилень у поведінці IoT–пристроїв. Саме на цьому рівні відбувається первинний збір і попередній аналіз даних. Для реалізації було використано мікроконтролер ESP8266 з підключеним до нього цифровим температурним датчиком DS18B20.

Вимірювання температури проводяться систематично – кожні пів хвилини. Для відстеження змін у роботі приладу, контролер не просто відправляє кожне значення на сервер, а створює локальний профіль – перелік з останніх 10

показників температури. Цей профіль дає змогу контролеру незалежно обчислювати середнє значення, а також визначати, чи поточний показник виходить за межі норми (наприклад, ±10% від середнього).

Алгоритм роботи контролера складається з наступних кроків:

– зчитування температури з датчика;

– додавання нового значення у список останніх 10 вимірів. якщо список вже повний, то найстаріше значення видаляється;

– обчислення середньої температури на основі буфера останніх даних;

– порівняння нового значення з нормою, якщо воно виходить за межі ±10%, визначається як аномалія;

– формування повідомлення: у вигляді json–структури, що містить температуру та статус («ok» або «anomaly»);

– передача даних на сервер за допомогою http post–запиту через wi–fi–з'єднання.

Контролер працює незалежно і не потребує постійного з'єднання з сервером. Якщо зв'язок тимчасово відсутній, пристрій може повторювати спроби відправлення даних або продовжувати локальний аналіз до моменту відновлення комунікації.

Для реалізації програмної логіки на мікроконтролері було використано Arduino IDE, оскільки вона має простий інтерфейс, підтримує ESP8266, і дозволяє легко підключати необхідні бібліотеки. Основними бібліотеками, що використовувалися, є:

– ESP8266WiFi.h – для підключення до Wi–Fi;

– OneWire.h і DallasTemperature.h – для зчитування даних з термометра;

– ArduinoJson.h (опційно) – для формування JSON–структур.

Контролер зберігає останні 10 значень температури у масиві. Це дозволяє не просто реагувати на одне випадкове відхилення, а оцінювати загальну тенденцію. Наприклад, якщо температура різко зростає і кілька вимірів поспіль

виходять за межі, це є більш надійним індикатором аномалії, ніж одиночне коливання.

Звідси випливає що приклда поведінки виглядатиме наступим чином:

– пристрій протягом 10 хвилин передає стабільну температуру близько 24°C;

– нове значення – 26.8°C. Середнє: 24.1°C. Допустимий діапазон (±10%): 21.7–26.5°C;

– 26.8°C > 26.5°C – зафіксовано аномалію;

– контролер позначає статус як «anomaly» і надсилає повідомлення на сервер.

3.4 Реалізація серверної частини

Серверний компонент розробленої системи здійснює централізований прийом, обробку та реагування на інформацію, що надходить від контролерів. Головна мета сервера – виявити нештатні зміни в роботі пристрою, зареєструвати подію, сповістити користувача або вжити інших заходів безпеки. Серверна частина логіки розроблена з урахуванням простоти впровадження та можливості подальшого масштабування. Для створення сервера було обрано мову Python, через її простоту в освоєнні, наявність великої кількості готових бібліотек і активне використання в сфері обробки даних та мережевих додатків. Основна логіка сервера реалізована з використанням мікрофреймворку Flask, що дозволяє швидко створити веб–сервер за мінімальної кількості коду. Цей сервер обробляє HTTP POST–запити з даними від ESP8266 і виконує перевірку на аномалії.

Основні етапи роботи серверної частини:

– запуск flask–сервера, який слухає вхідні з'єднання на порту (наприклад, 5000);

– отримання json–даних у тілі http–запиту, дані включають температуру та статус, який надіслав контролер;

– розбір та валідація отриманих даних, перевірка їх формату, наявності ключових полів (temperature, status);

– порівняння з нормою, якщо контролер надсилає лише «сирі» значення, сервер сам перевіряє їх відповідність нормі, використовуючи заздалегідь задані межі (наприклад, 20–26°c);

– реакція на аномалію, якщо значення температури виходить за межі або статус дорівнює «anomaly», сервер виконує відповідні дії;

– створення запису в журналі подій, який може бути збережений у текстовому файлі, базі даних або просто виведений у консоль;

– сервер надсилає підтвердження отримання та, за потреби, додаткові інструкції.

У базовому варіанті сервер перевіряє, чи входить отримане значення температури у заданий діапазон. Якщо температура виходить за межі, фіксується аномалія. Наприклад, для норми 20–26°C:

– отримано 24.1°C – все в межах, лог «OK»;

– отримано 27.8°C – аномалія, запис у лог з позначкою «ANOMALY».

У разі виявлення загрози система може також:

– створити тривожне повідомлення у консоль,

– зберегти інцидент у файл із міткою часу,

– надіслати сповіщення адміністратору (у розширеній версії).

Для збереження інформації про аномалії використовується журнал подій у вигляді звичайного текстового файлу (наприклад, log.txt) (рис 3.4), у якому кожен запис має формат [Дата] Тип події: Temperature = [температура в цельсіях] (Device: esp8266–1).

```
[2025-05-08 07:10:15] OK: Temperature = 22.1°C (Device: esp8266-1)
[2025-05-08 07:20:33] OK: Temperature = 25.4°C (Device: esp8266-1)
[2025-05-08 07:30:45] ANOMALY: Temperature = 19.7°C (Device: esp8266-1)
[2025-05-08 07:40:52] OK: Temperature = 20.0°C (Device: esp8266-1)
[2025-05-08 07:51:00] ANOMALY: Temperature = 26.8°C (Device: esp8266-1)
[2025-05-08 08:01:22] OK: Temperature = 23.5°C (Device: esp8266-1)
[2025-05-08 08:12:10] OK: Temperature = 24.6°C (Device: esp8266-1)
[2025-05-08 08:22:37] ANOMALY: Temperature = 27.3°C (Device: esp8266-1)
[2025-05-08 08:32:44] OK: Temperature = 21.3°C (Device: esp8266-1)
[2025-05-08 08:42:58] OK: Temperature = 26.0°C (Device: esp8266-1)
[2025-05-08 08:53:07] ANOMALY: Temperature = 18.9°C (Device: esp8266-1)
[2025-05-08 09:03:18] OK: Temperature = 22.8°C (Device: esp8266-1)
[2025-05-08 09:13:31] OK: Temperature = 23.9°C (Device: esp8266-1)
[2025-05-08 09:23:42] ANOMALY: Temperature = 27.1°C (Device: esp8266-1)
[2025-05-08 09:34:00] OK: Temperature = 24.2°C (Device: esp8266-1)
[2025-05-08 09:44:15] OK: Temperature = 20.5°C (Device: esp8266-1)
[2025-05-08 09:54:26] OK: Temperature = 25.8°C (Device: esp8266-1)
[2025-05-08 10:04:33] ANOMALY: Temperature = 26.5°C (Device: esp8266-1)
[2025-05-08 10:14:47] OK: Temperature = 23.1°C (Device: esp8266-1)
[2025-05-08 10:25:00] ANOMALY: Temperature = 19.5°C (Device: esp8266-1)
[2025-05-08 10:35:17] OK: Temperature = 24.9°C (Device: esp8266-1)
[2025-05-08 10:45:28] ANOMALY: Temperature = 28.2°C (Device: esp8266-1)
[2025-05-08 10:55:40] OK: Temperature = 22.2°C (Device: esp8266-1)
[2025-05-08 11:05:53] OK: Temperature = 20.9°C (Device: esp8266-1)
[2025-05-08 11:16:08] ANOMALY: Temperature = 18.3°C (Device: esp8266-1)
[2025-05-08 11:26:23] OK: Temperature = 23.7°C (Device: esp8266-1)
[2025-05-08 11:36:39] ANOMALY: Temperature = 26.9°C (Device: esp8266-1)
[2025-05-08 11:46:50] OK: Temperature = 25.2°C (Device: esp8266-1)
```

Рисунок 3.4 – Дані з датчика температури

Такий формат зручний для перегляду, архівування та подальшого аналізу. У більш складних реалізаціях можливо підключити базу даних (наприклад, SQLite або MongoDB), але в межах цієї роботи пріоритетом була простота.

3.5 Тестування роботи системи

Для контролю функціональності створеної системи було здійснено експериментальне випробування в локальному оточенні, що максимально повторює реальне. Завданням було не просто спостерігати за транспортуванням даних, а й візуально пересвідчитися, що система здатна розпізнавати аномалії в функціонуванні IoT–пристрою, оперативно реагувати на відхилення, відображати результати у доступному вигляді, а також зберігати стійкість під час довготривалого навантаження.

Весь експеримент починався зі старту контролера на базі ESP8266, підключеного до цифрового температурного сенсора DS18B20. Відразу після ввімкнення пристрій приєднався до локальної Wi–Fi мережі та почав зчитувати температуру навколишнього середовища з частотою один раз на 30 секунд. Усі зчитані значення зберігались у буфері з десяти останніх вимірів. Це дозволяло мікроконтролеру не лише бачити поточну температуру, а й самостійно аналізувати, чи не виходить вона за допустимі межі, визначені як ±10% від обчисленого середнього значення. Таким чином, ще до надсилання даних на сервер, пристрій вже виконував елементи поведінкового аналізу, що значно розвантажує центральну частину системи.

Протягом перших кількох хвилин досліду температура залишалася сталою. Усі показники коливалися в діапазоні від 23.8°C до 24.2°C. Контролер фіксував відповідність усіх даних нормі, формував пакети зі статусом «ok» та відправляв їх на сервер через HTTP POST–запити. Сервер, розроблений на Python з використанням Flask, отримував кожний запит, заносив інформацію до оперативного списку подій, відображав її у вигляді таблиці у веб–інтерфейсі та додатково записував у файл журналу. Таблиця в браузері була простою, але інформативною: кожний рядок містив час події, значення температури та статус пристрою. За звичайних обставин усі рядки мали зелений фон – як візуальне підтвердження контрольованості ситуації.

На наступному етапі тестування було зумисно створено аномальну ситуацію: сенсор було злегка нагріто пальцями руки, і вже наступне зчитування показало стрибок температури до 28.1°C. Контролер, порівнявши це значення із середнім у буфері (близько 24.0°C), виявив, що воно виходить за межі допустимого діапазону, й одразу позначив його як «anomaly». JSON–дані з відповідним статусом були відправлені на сервер, який відреагував миттєво. У таблиці браузера з’явився новий рядок з червоним фоном та позначкою аномалії. Одночасно в лог–файлі зафіксувалася подія з точною часовою міткою. Візуально

це виглядало переконливо: зелений потік даних був раптово перерваний червоним сигналом – чітким маркером потенційної загрози на рисунку 3.5.

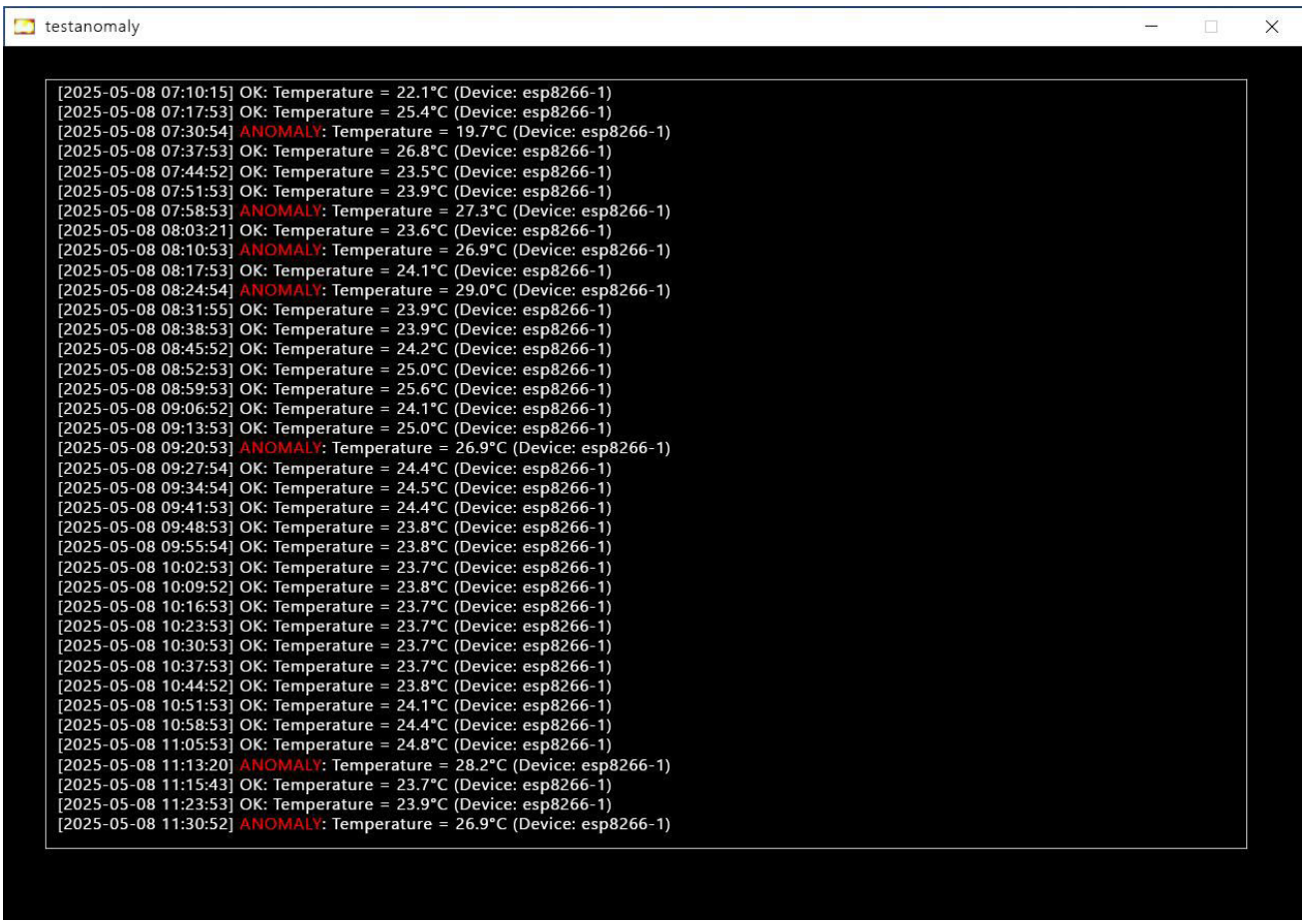


Рисунок 3.5 – Результат виявлення аномалій

Протягом кількох хвилин датчик продовжував нагріватися під час тесту, а всі наступні вимірювання були за межами допустимих значень. Сервер зафіксував низку сигналів тривоги, що свідчило про коректне функціонування контролера та алгоритму виявлення. Суттєво, що після припинення впливу температура повернулася до норми, і це було автоматично визначено контролером. Наступне повідомлення вже містило статус «ok», і система повернулася до нормального режиму роботи. Таблиця в браузері знову почала заповнюватися зеленими рядками.

Для перевірки стійкості було проведено ще один експеримент. Flask–сервер на кілька хвилин було зупинено. У цей час контролер не припиняв свою роботу –

він намагався передати дані, але не отримував відповіді. Після поновлення серверної частини зв’язок автоматично відновився без необхідності перезавантаження пристрою. Це показало, що система здатна витримувати короткочасні перебої в мережі без втрати стабільності або потреби в ручному втручанні.

Ще однією важливою складовою була зручність та візуалізація інтерфейсу. Веб–сторінка, що представляла поточний стан системи, не потребувала додаткового встановлення – працювала без проблем у будь–якому браузері. Кольорова індикація забезпечувала миттєве розпізнавання критичних змін. Статус «ok» виділявся зеленим, а «anomaly» – червоним кольором. Таблиця автоматично оновлювалася без перезавантаження сторінки, що забезпечувало комфортний і оперативний моніторинг.

Загалом, уся поведінка системи в процесі тестування підтвердила, що обрана архітектура – з передачею частини аналітики на контролер, простим текстовим обміном та базовою серверною логікою – цілком ефективна навіть у такій спрощеній реалізації. Виявлення аномалій працює точно, система не втрачає дані при збої, здатна самостійно відновлюватися, а користувач має змогу легко стежити за ситуацією через вебінтерфейс.

ВИСНОВКИ

У рамках цієї дипломної роботи проведено всебічне дослідження питання виявлення загроз у середовищі Інтернету речей (IoT), використовуючи поведінковий аналіз. Актуальність цієї теми визначається швидким збільшенням кількості IoT–пристроїв у різноманітних областях – від повсякденного життя до критичної інфраструктури – що, своєю чергою, призводить до зростання ризиків, пов’язаних із безпекою даних, стабільністю роботи систем та потенційними зловмисними діями. У теоретичній частині було детально проаналізовано особливості IoT–середовища, серед яких ключовими є обмежені обчислювальні ресурси, нестандартність платформ, гетерогенність протоколів зв’язку та обмежений або взагалі відсутній рівень безпеки у багатьох реалізаціях. Було показано, що традиційні методи захисту, такі як сигнатурне виявлення, міжмережеві екрани чи антивірусне ПЗ, не є ефективними в умовах IoT, де переважна частина пристроїв працює автономно, не маючи постійного підключення до централізованої системи оновлень або захисту. Отож, зосередженість у дослідженні була на підході до виявлення небезпек, що ґрунтується на аналізі дій. Сенс такого підходу в тому, що система досліджує типову (нормальну) поведінку обладнання в мережі, створює відповідний профіль і надалі слідкує за відхиленнями від цього, як за потенційними показниками атаки або збоїв. Цей спосіб дає можливість знаходити навіть ті загрози, про які раніше не знали, і які неможливо зафіксувати за допомогою традиційних методів виявлення на основі правил або сигнатур.

У ході роботи було сформовано чітке бачення архітектури системи, яка здатна працювати в умовах обмеженого доступу до ресурсів. Архітектура включає контролерну частину, розміщену безпосередньо на IoT–пристрої, та серверну частину, яка здійснює централізований аналіз, фіксацію та реагування. Була розроблена і реалізована модель, де роль IoT–пристрою виконував мікроконтролер ESP8266 із температурним сенсором DS18B20. Така

конфігурація була обрана як типова для простих пристроїв моніторингу середовища – наприклад, у серверних приміщеннях, дата–центрах, smart home–системах тощо. Найбільшу вартість у цій праці становить впровадження контролю за функціонуванням пристрою не тільки на сервері, а й безпосередньо у контролері. Завдяки цьому було презентовано концепцію розподіленого аналізу поведінки, де кожен прилад бере участь у забезпеченні власної безпеки. Це сприяє зменшенню навантаження на мережу загалом, прискоренню ідентифікації загроз і підвищенню самостійності системи.

У практичній частині було здійснено повну реалізацію всієї логіки: від збору даних до формування журналу подій, а також візуального представлення інформації для користувача у вигляді простого, але функціонального вебінтерфейсу. Система пройшла серію тестувань, у ході яких були змодельовані як нормальні, так і аномальні ситуації. Виявлення загроз на основі поведінкових відхилень відбувалося точно та своєчасно, про що свідчать результати журналів і реакція інтерфейсу в реальному часі. Була підтверджена здатність системи не лише виявляти разові відхилення, а й ефективно реагувати на тривалі аномалії, а також відновлювати нормальну роботу після зникнення загроз без потреби в ручному втручанні. Окрім того, здійснене випробування дало змогу упевнитися у стабільності системи у випадку непередбачених обставин, зокрема, у випадках тимчасової втрати з'єднання з сервером або коливаннях величин поблизу граничних значень. Контролерна частина системи в таких умовах продовжувала працювати без проблем, зберігала логіку поведінкового аналізу на місці та здійснювала повторні спроби передавання даних, що демонструє її здатність до самовідновлення. Це надзвичайно суттєво, коли надійність IoT–інфраструктури має ключове значення, наприклад, у системах безпеки, управління кліматом або технічному моніторингу. Серверна частина, реалізована з використанням Flask, також показала себе як надійна і гнучка основа для обробки подій. Вона не лише виконувала роль приймача інформації, а й активно аналізувала отримані дані, формувала реакцію у вигляді зміни статусу в інтерфейсі та записів у лог–файлах.

Завдяки простій структурі програмного коду та відкритій архітектурі, така система легко масштабована: до неї можна додавати нові типи пристроїв, вводити додаткові параметри (наприклад, вологість, тиск, споживання енергії), а також ускладнювати логіку поведінкового аналізу шляхом інтеграції з методами машинного навчання або статистичного моделювання.

Особливу увагу в дипломній праці було присвячено тому, аби логіка аналізу не була суворо прив’язана до певної апаратної платформи чи різновиду даних. Це дає змогу застосовувати розроблену архітектуру в різних ситуаціях. Приміром, пристрої, що зчитують електричне навантаження, вібрації, рівень шуму або руху, можуть так само бути додані до системи з невеликими змінами в коді. Поведінковий підхід, на відміну від сигнатурного, не залежить від конкретного різновиду загрози – він фіксує будь–яке нетипове відхилення, що й робить його універсальним.

Система продемонструвала високу адаптивність. Завдяки можливості самостійно обчислювати норму на основі буфера попередніх значень, вона не потребує жорстко прописаних правил чи налаштувань на кожен окремий випадок. При зміні середовища або характеру роботи пристрою, профіль поведінки автоматично оновлюється, що дозволяє уникати хибних спрацювань і водночас підтримувати високу чутливість до справжніх загроз.

Отже, дипломна робота реалізує не тільки технічний, а й концептуальний підхід до створення безпечного середовища для IoT, де кожен прилад здатний брати участь у забезпеченні цілісності системи. Втілення поведінкового аналізу навіть у спрощеному вигляді засвідчила свою ефективність, а збудована архітектура – свою життєздатність та практичну цінність.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. IoT Architecture: Definition, Explanation, and Use Cases Vation Ventures. Vation Ventures Innovation Starts Here. URL: https://www.vationventures.com/glossary/iot-architecture-definition-explanation-and-use-cases (дата звернення: 02.01.2025).

2. Top IoT Device Vulnerabilities: How To Secure IoT Devices Fortinet. Fortinet. URL: https://www.fortinet.com/resources/cyberglossary/iot-device-vulnerabilities (дата звернення: 02.01.2025).

3. Damor D. IoT Security Threats and Solutions. Einfochips. URL: https://www.einfochips.com/blog/iot-security-threats-and-solutions (дата звернення: 02.01.2025).

4. Romanets I. Що таке спуфінг і як запобігти атаці?. Навчально-науковий центр інформаційних технологій. URL: https://nncit.wunu.edu.ua/shho-take-spufing-i-yak-zapobigty-ataczi/ (дата звернення: 02.01.2025).

5. Що таке NDP (Neighbor Discovery Protocol)?. Форум Ruby для початківців - вивчаємо Рубі разом!. URL: https://rubydevelopers.org/t/ndp-neighbor-discovery-protocol/367 (дата звернення: 02.01.2025).

6. Що таке DDoS-атака?. Microsoft. URL: https://www.microsoft.com/uk-ua/security/business/security-101/what-is-a-ddos-attack (дата звернення: 02.01.2025).

7. IoT Security: Protecting Your Connected Devices. Nexusgroup. URL: https://www.nexusgroup.com/iot-security/ (дата звернення: 02.01.2025).

8. Перехоплення сеансу. CQR. URL: https://cqr.company/ua/web-vulnerabilities/session-hijacking/ (дата звернення: 02.01.2025).

9. Amod F. What are Internet of Things (IoT) attacks?. HIPAA Compliant Email Start For Free Encrypted Email by Paubox. URL: https://www.paubox.com/blog/what-are-internet-of-things-iot-attacks (дата звернення: 02.01.2025)

10. Vesna A. K., Gacovski Z. Methods for Detection and Prevention of Vulnerabilities in the IoT (Internet of Things) Systems. IntechOpen - Open Science

Open Minds IntechOpen. URL: https://www.intechopen.com/chapters/88775 (дата звернення: 02.01.2025)

11. Ghosh R., Vasan A., Puthal D., Mohanty S. Internet of Things Security: A Survey on Challenges, Attacks and Solutions. Springer. URL: https://link.springer.com/article/10.1007/s00500-020-04819-z (дата звернення: 02.01.2025).

12. IoT Security Risks. Kaspersky. [Електронний ресурс] – Режим доступу - URL: https://www.kaspersky.com/resource-center/threats/internet-of-things-risks (дата звернення: 02.01.2025).

13. Akhunzada A., Gani A., Khan M. K. et al. Secure and dependable service management for mobile cloud computing: a survey. Journal of Network and Computer Applications.URL:https://www.researchgate.net/publication/256181993_A_Survey_of_Mobile_Cloud_Computing_Application_Models (дата звернення: 02.01.2025).

14. Arduino Nano Documentation. Arduino.cc. URL: https://docs.arduino.cc/hardware/nano (дата звернення: 02.01.2025).

15. Dallas Temperature Control Library Documentation. Arduino Playground. URL: https://playground.arduino.cc/Learning/OneWire/ (дата звернення: 02.01.2025).

16. Flask Web Framework Documentation. Pallets Projects. URL: https://flask.palletsprojects.com/en/2.3.x/ (дата звернення: 03.01.2025).

17. Wi-Fi Module ESP8266 Technical Specifications. Espressif Systems. URL: https://www.espressif.com/en/products/socs/esp8266 (дата звернення: 03.01.2025).

18. OWASP Internet of Things Project. OWASP Foundation. URL: https://owasp.org/www-project-internet-of-things/ (дата звернення: 03.01.2025).

19. Cimpanu C. Microsoft: 83% of IoT Devices Are Vulnerable to Attack. ZDNet. URL: https://www.zdnet.com/article/microsoft-83-of-iot-devices-are-vulnerable-to-attack/ (дата звернення: 03.01.2025).

20. Mavropoulos O. IoT Devices and Firmware Vulnerabilities. Pentestmag. URL: https://pentestmag.com/iot-devices-and-firmware-vulnerabilities/ (дата звернення: 10.01.2025).

21. Mosenia A., Jha N. K. A comprehensive study of security of Internet-of-Things. IEEE Transactions on Emerging Topics in Computing. URL: https://ieeexplore.ieee.org/document/7413782 (дата звернення: 15.01.2025).

22. Farooq M. U., Waseem M., Mazhar S. A review on Internet of Things (IoT). International Journal of Computer Applications. URL: https://ijcaonline.org/archives/volume113/number1/19829-2015905125 (дата звернення: 18.01.2025).

23. Arduino JSON Library. Arduino.cc. URL: https://arduinojson.org/ (дата звернення: 23.01.2025).

24. REST API Concepts. RESTfulAPI.net. URL: https://restfulapi.net/ (дата звернення: 25.01.2025).

25. Sensors in IoT. TutorialsPoint. URL: https://www.tutorialspoint.com/internet_of_things/internet_of_things_sensors.htm (дата звернення: 05.02.2025).

26. Ray P. A survey on Internet of Things architectures. Journal of King Saud University – Computer and Information Sciences. URL: https://www.sciencedirect.com/science/article/abs/pii/S1319157817301039 (дата звернення: 10.02.2025).

27. Humayed A., Lin J., Li F., Luo B. Cyber-physical systems security - A survey. IEEE Internet of Things Journal. URL: https://ieeexplore.ieee.org/document/7845487 (дата звернення: 18.02.2025).

28. Kasinathan P., Pastrone C., Spirito M., Vinkovits M. Denial-of-Service detection in 6LoWPAN based Internet of Things. IEEE. URL: https://ieeexplore.ieee.org/document/6496935 (дата звернення: 18.02.2025).

29. Alrawais A., Alhothaily A., Hu C., Cheng X. Fog Computing for the Internet of Things: Security and Privacy Issues. IEEE Internet Computing. URL: https://ieeexplore.ieee.org/document/7926928 (дата звернення: 25.02.2025).

30. Islam S. R., Kwak D., Kabir M. H., Hossain M., Kwak K. S. The Internet of Things for Health Care: A Comprehensive Survey. IEEE Access URL: https://ieeexplore.ieee.org/document/7113786 (дата звернення: 27.02.2025).

31. Cervone H. F. Understanding Cybersecurity Management in Modern Organizations. IGI Global. URL: https://www.igi-global.com/book/understanding-cybersecurity-management-modern-organizations/275763 (дата звернення: 03.03.2025).

32. Weerasinghe S., Halgamuge S. Behavioural Pattern Analysis for IoT Threat Detection. MDPI Sensors. URL: https://www.mdpi.com/1424-8220/20/7/1975 (дата звернення: 03.03.2025).

33. Roman R., Najera P., Lopez J. Securing the Internet of Things. Computer. IEEE. URL: https://ieeexplore.ieee.org/document/6025320 (дата звернення: 06.03.2025).

34. Dorsemaine B., Gaulier J. P., Wary J. P., Kheir N., Urien P. Internet of Things: A Definition and Taxonomy. In: Next Generation Mobile Applications, Services and Technologies. IEEE. URL: https://ieeexplore.ieee.org/document/7317398 (дата звернення: 10.03.2025).

35. Sicari S., Rizzardi A., Grieco L. A., Coen-Porisini A. Security, privacy and trust in Internet of Things: The road ahead. Computer Networks. URL: https://www.sciencedirect.com/science/article/pii/S1389128614002843 (дата звернення: 15.03.2025).

36. Kumar R., Mallick P. K. The Internet of Things: Insights into the building blocks, component interactions, and architecture layers. Procedia Computer Science. URL: https://www.sciencedirect.com/science/article/pii/S1877050915008538 (дата звернення: 15.03.2025).

37. Siboni S., Sachidananda V., Meidan Y., Bohadana M., Mathov Y., Mirsky Y., Elovici Y. Security Testbed for Internet-of-Things Devices. ACM Transactions on Internet Technology. URL: https://dl.acm.org/doi/10.1145/3318144 (дата звернення: 15.03.2025).

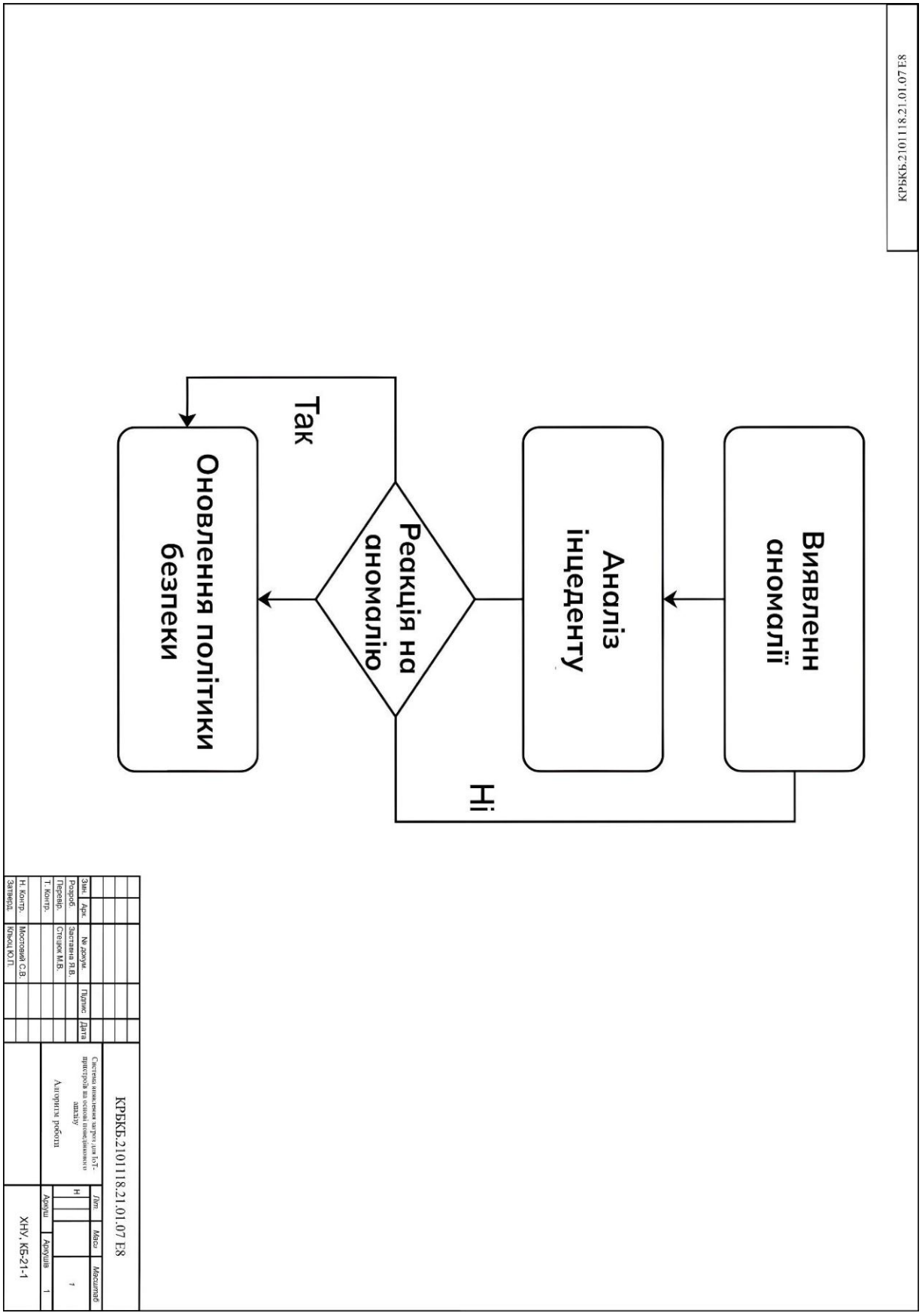
38. Fernandes E., Jung J., Prakash A. Security Analysis of Emerging Smart Home Applications. IEEE Symposium on Security and Privacy. URL: https://ieeexplore.ieee.org/document/7163049 (дата звернення: 20.03.2025).

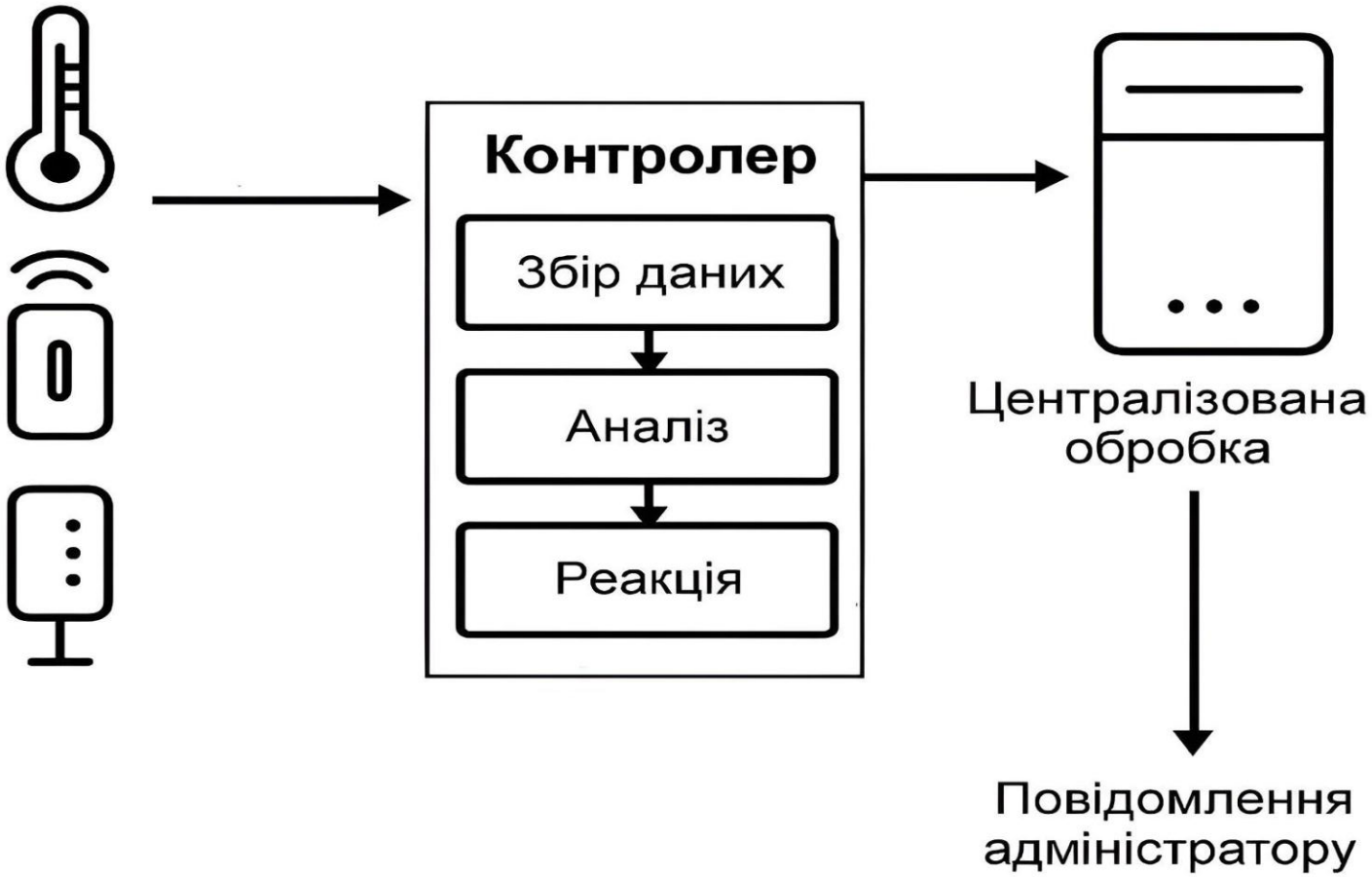
39. Sicari S., Rizzardi A., Grieco L. A., Coen-Porisini A. A survey on the security of the Internet of Things: Current status and open issues. Computer Communications. URL: https://www.sciencedirect.com/science/article/pii/S014036641400409X (дата звернення: 20.03.2025).

40. Lee I., Lee K. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons. URL: https://www.sciencedirect.com/science/article/abs/pii/S0007681315000373 (дата звернення: 20.03.2025).

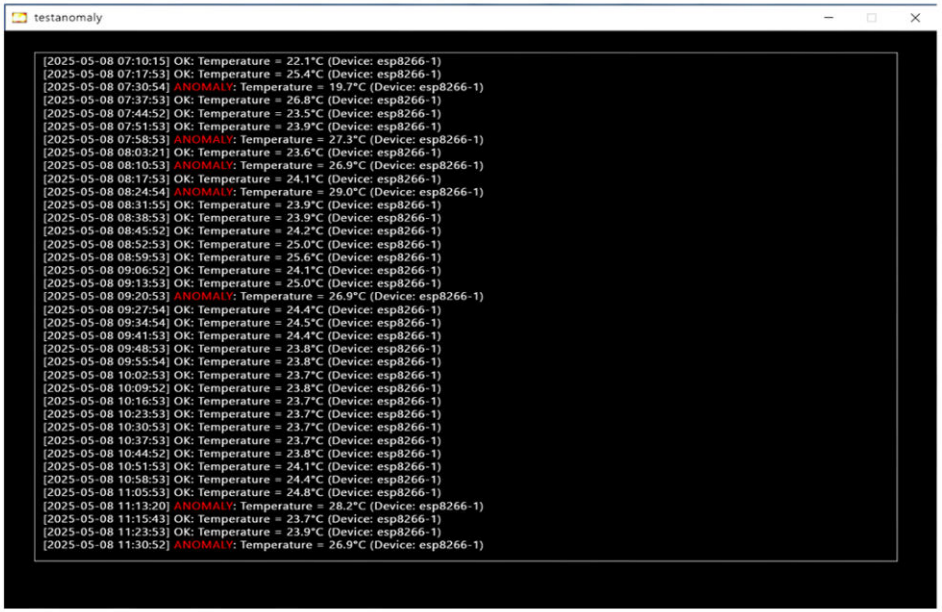
ДОДАТОК А

Копії графічної частини





					КРБКБ.2101118.21.01.07 Е2			
					Система виявлення загроз для IoT-пристроїв на основі поведінкового аналізу	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата		Н		1
Розроб.		Заставна Я.В.			Функціональна схема	Аркуш	Аркушів	1
Перевір.		Стецюк М.В.						
Т. Контр.								
Н. Контр.		Мостовий С.В.				ХНУ, КБ-21-1		
Затверд.		Кльоц Ю.П.						




```
[2025-05-08 07:10:15] OK: Temperature = 22.1°C (Device: esp8266-1)
[2025-05-08 07:20:33] OK: Temperature = 25.4°C (Device: esp8266-1)
[2025-05-08 07:30:45] ANOMALY: Temperature = 19.7°C (Device: esp8266-1)
[2025-05-08 07:40:52] OK: Temperature = 20.0°C (Device: esp8266-1)
[2025-05-08 07:51:00] ANOMALY: Temperature = 26.8°C (Device: esp8266-1)
[2025-05-08 08:01:22] OK: Temperature = 23.5°C (Device: esp8266-1)
[2025-05-08 08:12:10] OK: Temperature = 24.6°C (Device: esp8266-1)
[2025-05-08 08:22:37] ANOMALY: Temperature = 27.3°C (Device: esp8266-1)
[2025-05-08 08:32:44] OK: Temperature = 21.3°C (Device: esp8266-1)
[2025-05-08 08:42:58] OK: Temperature = 26.0°C (Device: esp8266-1)
[2025-05-08 08:53:07] ANOMALY: Temperature = 18.9°C (Device: esp8266-1)
[2025-05-08 09:03:18] OK: Temperature = 22.8°C (Device: esp8266-1)
[2025-05-08 09:13:31] OK: Temperature = 23.9°C (Device: esp8266-1)
[2025-05-08 09:23:42] ANOMALY: Temperature = 27.1°C (Device: esp8266-1)
[2025-05-08 09:34:00] OK: Temperature = 24.2°C (Device: esp8266-1)
[2025-05-08 09:44:15] OK: Temperature = 20.5°C (Device: esp8266-1)
[2025-05-08 09:54:26] OK: Temperature = 25.8°C (Device: esp8266-1)
[2025-05-08 10:04:33] ANOMALY: Temperature = 26.5°C (Device: esp8266-1)
[2025-05-08 10:14:47] OK: Temperature = 23.1°C (Device: esp8266-1)
[2025-05-08 10:25:00] ANOMALY: Temperature = 19.5°C (Device: esp8266-1)
[2025-05-08 10:35:17] OK: Temperature = 24.9°C (Device: esp8266-1)
[2025-05-08 10:45:28] ANOMALY: Temperature = 28.2°C (Device: esp8266-1)
[2025-05-08 10:55:40] OK: Temperature = 22.2°C (Device: esp8266-1)
[2025-05-08 11:05:53] OK: Temperature = 20.9°C (Device: esp8266-1)
[2025-05-08 11:16:08] ANOMALY: Temperature = 18.3°C (Device: esp8266-1)
[2025-05-08 11:26:23] OK: Temperature = 23.7°C (Device: esp8266-1)
[2025-05-08 11:36:39] ANOMALY: Temperature = 26.9°C (Device: esp8266-1)
[2025-05-08 11:46:50] OK: Temperature = 25.2°C (Device: esp8266-1)
```

					КРБКБ.2101118.21.01.07 Е8			
					Система виявлення та захисту від DoS-атак в корпоративній мережі	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата		Н		1
Розроб.		Заставна Я.В.			Тестування програми	Аркуш	Аркушів	1
Перевір.		Стецюк М.В.						
Т. Контр.						ХНУ, КБ-21-1		
Н. Контр.		Мостовий С.В.						
Затверд.		Кльоц Ю.П.						

ДОДАТОК Б

Програмний код

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <ArduinoJson.h>

// Wi-Fi налаштування
const char* ssid = "MyHomeWiFi";
const char* password = "securepass123";

// Адреса сервера для надсилання даних
const char* serverUrl = "http://192.168.1.100:5000/temperature";

// Пін, до якого підключено DS18B20
#define ONE_WIRE_BUS D4  // GPIO2

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

// Буфер останніх 10 значень температури
float tempBuffer[10] = {25.0};  // Ініціалізовано з 25.0°C
int bufferIndex = 0;

void setup() {
  Serial.begin(115200);
  sensors.begin();
```

```
  WiFi.begin(ssid, password);
  Serial.print("Підключення до Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWi-Fi підключено");
}

void loop() {
  sensors.requestTemperatures();
  float currentTemp = sensors.getTempCByIndex(0);
  Serial.print("Температура: ");
  Serial.println(currentTemp);

  // Додавання в буфер
  tempBuffer[bufferIndex] = currentTemp;
  bufferIndex = (bufferIndex + 1) % 10;

  // Розрахунок середнього значення
  float avgTemp = 0;
  for (int i = 0; i < 10; i++) {
    avgTemp += tempBuffer[i];
  }
  avgTemp /= 10;

  // Визначення статусу
  String status = "ok";
  if (abs(currentTemp - avgTemp) > avgTemp * 0.10) {
    status = "anomaly";
```

```
  }

  // Вивід у консоль
  Serial.print("Середня температура: ");
  Serial.println(avgTemp);
  Serial.print("Статус: ");
  Serial.println(status);

  // Надсилання даних через HTTP POST
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    http.begin(serverUrl);
    http.addHeader("Content-Type", "application/json");

    // Формування JSON
    StaticJsonDocument<200> jsonDoc;
    jsonDoc["temperature"] = currentTemp;
    jsonDoc["average"] = avgTemp;
    jsonDoc["status"] = status;
    String jsonStr;
    serializeJson(jsonDoc, jsonStr);

    int httpResponseCode = http.POST(jsonStr);
    Serial.print("HTTP статус: ");
    Serial.println(httpResponseCode);

    http.end();
  }
  delay(30000); // Затримка 30 секунд
}
```

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.

Заставної Яни Валентинівни

ПІБ здобувача вищої освіти

Студентки ФІТ, 4 курсу, групи КБ-21-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 31.08.2023, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлена. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщена та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

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

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

02.06.2025
дата

підпис

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 0.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 8%**

	Document		Sum coincidence on the DB	
ID: 243440 Title: Система виявлення загроз для IoT-пристроїв на основі поведінкового аналізу Added in a DB: 2025-06-04 Authors: Заставна Яна Валентинівна Heads: Стецюк М.В. Consultants: Opponents:	Symbols	Lexemes	Symbols	Lexemes
	90482	680	494 (1%)	6 (1%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності науковим керівником

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

Автор: Заставна Яна Валентинівна

Співавтор:

Назва: Система виявлення загроз для IoT-пристроїв на основі поведінкового аналізу

Науковий керівник:

Підрозділ: Кафедра кібербезпеки

Коефіцієнт подібності 1:1.9%

Коефіцієнт подібності 2:0%

Мікропробіли: 0

Заміна букв: 3

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-04 23:24:14.0

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:

05.06.2025р.

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

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

Назва: Система виявлення загроз для IoT-пристроїв на основі поведінкового аналізу.

Автор: Заставна Яна Валентинівна

Спеціальність: 125 – Кібербезпека

Освітня програма: Кібербезпека

Науковий керівник: Микола Стецюк, канд. техн. наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

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

Підтвердження:

Оригінальність тексту роботи за результатами перевірки системою StrikePlagiarism складає 98,1%, оригінальність тексту роботи за результатами перевірки системою Anti-Plagiarism складає 100%.

Згідно з правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 24.09.2024, авторська робота, обсяг оригінального тексту у відсотках до загального обсягу матеріалу в якій складає 90-100%, визначається роботою з високою унікальністю тексту і допускається до захисту.

Керівник роботи _____ Микола СТЕЦЮК

Гарант ОП _____ Віктор ЧЕШУН

Завідувач кафедри кібербезпеки _____ Юрій КЛЬОЦ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «бакалавр»

Студентка Заставна Яна Валентинівна

Тема Система виявлення загроз для IoT-пристроїв на основі поведінкового аналізу

Спеціальність 125 – Кібербезпека

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «бакалавр»:

кількість листів креслень 3; кількість сторінок записки 67.

1. Короткий зміст роботи та прийнятих рішень: У кваліфікаційній роботі розглянуто актуальні питання безпеки в мережах Інтернету речей (IoT). Проаналізовано архітектуру IoT-систем, їхню вразливість, види загроз і атак. В теоретичній частині представлено сучасні методи виявлення аномалій, зокрема поведінковий аналіз. У практичній частині спроєктовано систему виявлення загроз на основі поведінкових моделей пристроїв, реалізовано її програмний прототип, проведено тестування роботи системи.

2. Висновок про відповідність кваліфікаційної роботи завданню: Робота повністю відповідає поставленому завданню, містить як теоретичний аналіз проблеми, так і практичну реалізацію та тестування запропонованого рішення.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі висвітлено сучасні загрози безпеці IoT-пристроїв, класифікацію атак, уразливості архітектури та підходи до захисту. У другому розділі розроблено архітектуру системи виявлення загроз, описано її компоненти, принципи формування нормальної поведінки пристроїв та реагування на аномалії. Третій розділ присвячено реалізації програмного прототипу системи, вибору технологій та тестуванню роботи. У роботі використано сучасні підходи в сфері кібербезпеки, поведінкового аналізу та машинного навчання.

4. Позитивні сторони роботи: Актуальність теми, обґрунтованість вибору методів, чітка структура викладеного матеріалу. Значна увага приділена практичній реалізації та тестуванню, що підкреслює прикладну цінність роботи. Використано сучасні інструменти та підходи до моніторингу поведінки IoT-пристроїв.

5. Негативні сторони роботи: У деяких розділах відчувається надмірна деталізація загальновідомих теоретичних відомостей. Було б доцільно більше уваги приділити порівнянню запропонованої моделі з альтернативними методами виявлення загроз.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Графічні матеріали (схеми, ілюстрації) відповідають тематиці, охайні та зрозумілі. Пояснювальна записка оформлена відповідно до встановлених стандартів і вимог.

7. Відгук про роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки, оскільки матеріал викладено послідовно, логічно та з дотриманням вимог до оформлення. Усі розділи пов'язані між собою, добре розкривають теоретичні основи та демонструють практичне застосування розробленої системи. Робота відображає належний рівень підготовки студента та його здатність до самостійного розв'язання прикладних завдань у сфері кібербезпеки IoT-пристроїв.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи: Враховуючи високий рівень виконання, актуальність теми та практичну значущість, кваліфікаційна робота заслуговує оцінки **«відмінно»**.

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)
Бойко Юлій Миколайович, доктор технічних наук, професор кафедри ТМІТ

«06» 06 2025. _____ (підпис)