Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «МАГІСТР»

ДОСЛІДЖЕННЯ ТА АНАЛІЗ МОДУЛЮ БЕЗПЕКИ ІНФОРМАЦІЇ АРХІТЕКТУРИ IOT

RESEARCH AND ANALYSIS OF THE INFORMATION SECURITY MODULE OF THE IOT ARCHITECTURE

спеціальність 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

Виконав: здобувач вищої освіти
групи КНм-21
Столярук Максим Сергійович

_____
(підпис)

Керівник: к.т.н., доцент
Кошелюк Віктор Андрійович

_____
(підпис)

Кваліфікаційну роботу
допущено до захисту
«___» _____ 2025 р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Валерій Олександрович

_____
(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних та інформаційних технологій*
Кафедра *комп'ютерних наук*
Ступінь вищої освіти: *магістр*
Галузь знань: *12 Інформаційні технології*
Спеціальність: *122 Комп'ютерні науки*
Освітня програма: *«Комп'ютерні науки»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____Валерій ЛІЩИНА
«14» травня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА
ДРУГОГО (МАГІСТЕРСЬКОГО) РІВНЯ ВИЩОЇ ОСВІТИ

Столярук Максим Сергійович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Дослідження та аналіз модулю безпеки інформації архітектури IoT»
Керівник к.т.н., доцент Кошелюк Віктор Андрійович
затверджені наказом закладу вищої освіти від «*14*» *травня* *2025* р. № 255/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи *«05» грудня 2025 р.*

3. Вихідні дані до роботи: _____
_____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):
Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз і вибір засобів проектування, опис функціонального наповнення об'єкта проектування, розробка й обґрунтування системного наповнення, експерементальне дослідження результативності предмету дослідження.
_____

5. Перелік графічного матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз проблематики за темою роботи та постановка завдань дослідження	*Кошелюк В.А.*		
Теоретичне дослідження та практична реалізація предмету дослідження	*Кошелюк В.А.*		
Експерементальне дослідження результативності предмету дослідження	*Кошелюк В.А.*		
Показник запозичень тексту	_____ %		
Інструментальна перевірка	*Кошелюк В. А.*		
Нормоконтроль	*Сачук В. О.*		
Гарант ОПП	*Ліщина В. О.*		

7. Дата видачі завдання *«14» травня 2025 р.*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	*Провести огляд літературних джерел по темі кваліфікаційної роботи*	*до 30.06.2025 р*	
2	*Провести аналіз загальної проблеми і вибір напрямків дослідження*	*до 01.09.2025 р.*	
3	*Розробити функціональну схему роботи програмного продукту*	*до 01.10.2025 р*	
4	*Описати засоби розробки об'єкта проектування*	*до 15.10.2025 р.*	
5	*Практична реалізація об'єкта проектування*	*до 10.11.2025 р.*	
6	*Провести експерементальне дослідження результативності предмету дослідження*	*до 25.11.2025 р.*	
7	*Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру*	*до 05.12.2025 р.*	

Здобувач вищої освіти _____ Максим СТОЛЯРУК

Керівник роботи _____ Віктор КОШЕЛЮК

АНОТАЦІЯ

Столярук М. С. Дослідження та аналіз модулю безпеки інформації архітектури IoT. Рукопис. Кваліфікаційна робота магістра за спеціальністю 122 Комп’ютерні науки. Луцький національний технічний університет. Луцьк, 2025. 64 с.

Кваліфікаційна робота магістра складається з вступу, трьох розділів, висновків, списку використаних джерел, додатків.

У роботі досліджено та проведено аналіз модулю безпеки інформації архітектури IoT з використанням машинного навчання. Під час виконання поставлених завдань було проаналізовано методи та засоби виявлення загроз та вразливостей архітектури інтернету речей, досліджено технології управління інформаційною безпекою, здійснено конфігурацію та реалізацію модулю безпеки інформації архітектури IoT з використанням методу KNN.

Ключові слова: безпека, вразливості, загрози, машинне навчання, ELK.

ANNOTATION

Maksym Stolyaruk. Research and analysis of the information security module of the IoT architecture. Manuscript. Master’s Qualification Thesis in the field of 122 Computer Science. Lutsk National Technical University, 2025. 64 pages.

The master’s thesis consists of an introduction, three sections, conclusions, a list of used sources, appendices.

The paper investigates and analyzes the IoT architecture information security module using machine learning. During the implementation of the tasks, methods and means of detecting threats and vulnerabilities in the Internet of Things architecture were analyzed, information security management technologies were investigated, and the IoT architecture information security module was configured and implemented using the KNN method.

Keywords: security, vulnerabilities, threats, machine learning, ELK.

ЗМІСТ

ВСТУП…… 6
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ БЕЗПЕКИ ІОТ ТА ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ…..…… 10
1.1 Огляд і аналіз предметної області проблеми, результати існуючих теоретичних та експериментальних досліджень...…….. 10
1.2 Огляд і аналіз методів та засобів безпеки IoT для вирішення проблеми дослідження…….……..……..... 21
1.3 Постановка завдання на кваліфікаційну роботу магістра..…….. 28
Висновки до розділу 1...…… 29
РОЗДІЛ 2 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ МОДУЛЮ БЕЗПЕКИ ІНФОРМАЦІЇ……..... 30
2.1 Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання…… 30
2.2 Практична реалізація об’єкта проектування…….. 32
Висновки до розділу 2...…… 44
РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МОДУЛЮ БЕЗПЕКИ ІНФОРМАЦІЇ…….. 45
3.1 Методика проведення дослідження……... 45
3.2 Обробка та аналіз отриманих результатів.……... 51
Висновки до розділу 3…….…….. 59
ВИСНОВКИ…… 60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ…….. 63
ДОДАТКИ……... 65

ВСТУП

Актуальність дослідження. Поява Інтернету докорінно змінила людське життя. Завдяки здешевленню мікросхем і розвитку високошвидкісних телекомунікацій, Мережа дозволила об’єднувати пристрої між собою. Це відкрило шлях до того, що повсякденні речі – від зубних щіток і телевізорів до холодильників та автомобілів – можуть використовувати датчики для збору даних і розумно реагувати на дії користувачів. Саме ця глобальна мережа взаємопов’язаних пристроїв і отримала назву IoT (Інтернет речей).

Інтернет речей, попри свою новизну, вже глибоко інтегрувався у наше життя, трансформуючи взаємодію між користувачем та пристроями. Візьміть, наприклад, розумний будинок: тут техніка функціонує злагоджено – від автоматичного ввімкнення кавоварки та відкриття штор після спрацювання будильника, до самостійного замовлення продуктів холодильником.

Відкритість та недостатній захист більшості пристроїв IoT створюють серйозну вразливість до кіберзагроз. Оскільки ці системи часто не мають потужних механізмів безпеки, хакери можуть легко зловживати цими слабкостями, щоб викрасти особисті дані або навіть перебрати контроль над критично важливими інфраструктурами. Численні випадки несанкціонованого доступу до розумних будинків, автомобілів та промислових мереж чітко вказують на гостру потребу в розробці ефективних рішень для захисту IoT.

Ключ до цієї автоматизації – дані, які проходять багатоетапний шлях: збір, підготовка, обробка, перевірка та зберігання. Лише після цього ми отримуємо цінну інформацію. Однак, незахищеність цієї процедури робить її вразливою. У світі IoT, де обробляються критичні дані, це створює ідеальне поле для атак зловмисників. Отже, забезпечення безпеки шляхом розробки та обов’язкового використання систем моніторингу є нагальною потребою.

Інтернет речей кардинально змінює наш світ, але ця інновація супроводжується зростанням ризиків кібербезпеки. На жаль, у багатьох комерційних і промислових проєктах пристрої та інфраструктурні елементи IoT

швидко виявляють критичні вразливості. Це створює сприятливе поле для кіберзлочинців, які використовують прагнення зробити кожен пристрій «розумним». Зважаючи на потенційний вплив цих загроз на економіку, бізнес-операції, конфіденційність та безпеку, їхнє значення буде лише посилюватися. Аналітичні компанії у сфері безпеки роблять невтішні прогнози щодо ключових вразливостей. Деякі порушення можуть не тільки призвести до банкрутства компаній, але й завдати суттєвої шкоди звичайним громадянам.

Історично, багато сегментів та кінцевих пристроїв, на кшталт «розумної» побутової техніки (холодильники, пральні машини тощо), які зараз охоплені IoT, не мали класичних вимог до кібербезпеки. З огляду на інтенсивну конкуренцію, розробники цих продуктів зіткнулися з викликом. Вони не мають чіткого розуміння того, як здійснити безпечний життєвий цикл своїх пристроїв – від розробки та розгортання до інтеграції в IoT-інфраструктуру – з дотриманням усіх регуляторних норм.

Мета даної кваліфікаційної роботи магістра полягає у дослідженні та вивченні технологій керування інформаційною безпекою та побудові модуля безпеки інформації для IoT-систем із застосуванням машинного навчання.

Для досягнення визначеної мети потрібно виконати такі завдання:

– провести комплексне дослідження існуючого рівня безпеки інтернету речей (IoT), що охоплює дослідження протоколів передачі даних; аналіз архітектурних концепцій побудови IoT-систем; огляд існуючих методів та засобів захисту;

– розробити альтернативний модуль безпеки інформації архітектури IoT на базі Elastic Stack;

– експериментально перевірити роботу модулю безпеки інформації з використанням методів машинного навчання;

– розробити та впровадити програмне рішення, яке забезпечує комплексну обробку наборів даних, включаючи їх попередню підготовку, оптимізацію структури даних, процес навчання моделі та проведення тестування з метою валідації отриманих результатів.

– виконати аналіз отриманих експериментальних даних та результатів дослідження з метою формування конкретних, практично застосовних рекомендацій щодо підвищення рівня кібербезпеки IoT-екосистем.

Об’єктом дослідження є технології управління інформаційною безпекою та процеси моніторингу архітектури IoT на базі Elastic Stack.

Предметом даного дослідження є процеси виявлення та аналізу вразливостей у системах безпеки IoT-пристроїв. Особлива увага приділяється дослідженню сучасних методів тестування на проникнення, що застосовуються до таких пристроїв, оцінці їхньої стійкості до кіберзагроз, а також виявленню потенційних слабких місць. На основі отриманих результатів розробляються практичні рекомендації та заходи, спрямовані на підвищення рівня захищеності IoT-інфраструктури.

Наукова новизна дослідження полягає у створенні та обґрунтуванні модуля безпеки інформації для архітектури IoT на основі Elastic Stack із впровадженням алгоритмів машинного навчання. Запропонований підхід враховує обмежені обчислювальні ресурси IoT-пристроїв, характерні типи кіберзагроз і вимоги до ефективності виявлення аномалій. У роботі ґрунтовно проаналізовано рекомендації ENISA та перелік OWASP IoT Top 10, що використано для оцінки ризиків і моделювання потенційних вразливостей у сучасних IoT-системах.

Практична цінність проведеного дослідження полягає в можливості застосування розробленого модулю безпеки для комплексного аналізу та оцінювання рівня захищеності IoT-пристроїв, що функціонують у домашніх мережах. Отримані результати роботи створюють надійну основу для систематичної ідентифікації найпоширеніших вразливостей, включаючи незахищені мережеві сервіси, відкриті комунікаційні порти та некоректні конфігурації підключених пристроїв.

Здійснений ґрунтовний аналіз потенційних загроз дозволив сформувати практичні рекомендації щодо суттєвого підвищення рівня безпеки IoT-систем, які представляють значну практичну цінність як для звичайних користувачів домашніх мереж, так і для професійних фахівців у галузі кібербезпеки.

Запропонована методологія та інструментарій забезпечують можливість своєчасного виявлення потенційних загроз на ранніх стадіях їх формування, що дозволяє ефективно мінімізувати ризики несанкціонованого доступу до системи, витоку конфіденційних даних користувачів та інших видів кіберзлочинних атак, спрямованих на IoT-пристрої, підвищуючи загальний рівень інформаційної безпеки домашньої екосистеми.

Апробація результатів дослідження:

– 1 Міжнародна науково-практична конференція «The Integration of Research, Innovation and Economy» (08-10 жовтня 2025 р.), Севілья, Іспанія. International Science Group. 2025 [1].

– 1 Міжнародна науково-практична конференція «Modern Challenges in Economic and Technological Innovation» (15-17 жовтня 2025 р.), Болонья, Італія. International Science Group. 2025 [2].

РОЗДІЛ 1
АНАЛІЗ ПРОБЛЕМАТИКИ БЕЗПЕКИ ІОТ ТА ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ

1.1 Огляд і аналіз предметної області проблеми, результати існуючих теоретичних та експериментальних досліджень

Інтернет речей відкриває величезні можливості для бізнесу, тому дедалі більше компаній прагнуть інтегрувати ці інноваційні технології у свої робочі процеси. Однак перехід від теорії до практики супроводжується численними викликами та складнощами. Організації стикаються з проблемами масштабування, адже необхідно координувати роботу величезної кількості взаємопов’язаних пристроїв та забезпечити виконання специфічних технічних вимог для ефективної інтеграції всього виробничого ланцюга.

Щороку кількість підключених пристроїв збільшується з вражаючою швидкістю, демонструючи експоненціальне зростання. Провідні технологічні корпорації та наукові установи інтенсивно досліджують можливості впровадження інтернету речей у найрізноманітніші сфери людської діяльності. Особливо помітний прогрес спостерігається в медичній галузі, де IoT-пристрої поступово трансформують традиційні підходи до діагностування захворювань та терапевтичних процедур.

Окрім медичної сфери, технології Інтернету речей активно впроваджуються в аграрний сектор, де вони демонструють значний потенціал для модернізації традиційних методів господарювання. Сучасні фермерські господарства дедалі частіше застосовують інтелектуальні системи автоматизованого зрошення, які здатні оптимізувати витрати води залежно від потреб конкретних культур. Розумні сенсори безперервно відстежують параметри ґрунту, включаючи рівень вологості, кислотність, температуру та вміст поживних речовин, що дозволяє приймати обґрунтовані рішення щодо догляду за посівами. Завдяки цим інноваційним рішенням аграрії отримують можливість значно раціональніше використовувати водні та енергетичні

ресурси, а також точніше прогнозувати майбутні врожаї, що критично важливо для планування виробництва.

Отже, IoT-пристрої поступово трансформуються на невід'ємну складову як повсякденного побуту громадян, так і різноманітних економічних галузей. Значущість цих передових технологій невпинно зростає, перетворюючи їх на базис для майбутніх технологічних проривів і цифрової трансформації суспільства. Водночас масштабне поширення підключених пристроїв породжує серйозні виклики у сфері кібербезпеки, оскільки кожен новий девайс у глобальній мережі потенційно може стати вразливою точкою для несанкціонованого доступу. Враховуючи цю реальність, дослідження безпеки IoT-екосистем та створення надійних механізмів захисту набувають критичного значення, адже від рівня захищеності цих технологій залежить не лише економічна ефективність підприємств та якість повсякденного життя людей, але й безпосередньо життя та здоров'я окремих осіб.

Серед найбільш революційних розробок варто виділити проєкт Neuralink – амбітну ініціативу компанії під керівництвом відомого підприємця та новатора Ілона Маска. Ця технологія спрямована на створення прямого інтерфейсу між людським мозком та цифровими системами. Концепція передбачає зчитування електричних імпульсів мозку, їх інтелектуальну обробку та подальшу передачу до зовнішніх пристроїв або хмарних сервісів для глибокого аналізу [3].

Те, що ще недавно здавалося сюжетом фантастичних романів, сьогодні стає реальністю. Ця інноваційна технологія відкриває безпрецедентні перспективи для медицини та відновлювальної терапії. Насамперед, вона обіцяє кардинально змінити життя людей з фізичними обмеженнями, надаючи їм нові можливості для комунікації, руху та взаємодії з навколишнім світом, що раніше здавалося неможливим.

Сучасна IoT-екосистема являє собою складну мережеву інфраструктуру, що об'єднує різноманітні компоненти: розумні датчики, хмарні обчислювальні платформи, комунікаційні протоколи та багаторівневу архітектуру. Така

ієрархічна структура надає системним адміністраторам потужні інструменти для моніторингу, управління та забезпечення безперебійної роботи всієї системи.

Успішне впровадження IoT вимагає ретельного планування та проектування. Архітектура системи має органічно інтегруватися з існуючою IT-інфраструктурою підприємства, щоб забезпечити максимальну ефективність та віддачу від інвестицій. Фундаментом такої системи є багаторівнева архітектура IoT, що дозволяє контролювати технологічні процеси на всіх етапах. Критично важливо визначити та сформувати ці архітектурні рівні ще на початкових стадіях проектування, до розгортання мережевої інфраструктури. Класична IoT-архітектура зазвичай базується на трьох фундаментальних рівнях, кожен з яких виконує специфічні функції у загальній системі. На рисунку 1.1 проілюстровано рівні моделі OSI для IoT пристроїв.

Рисунок 1.1 – Рівні моделі OSI для IoT пристроїв [4]

Сучасна архітектура Інтернету речей (IoT) базується на трьох ключових рівнях, кожен з яких виконує специфічні функції для забезпечення ефективної роботи системи.

Рівень IoT-пристроїв представляє собою клієнто-орієнтований шар взаємодії, де відбувається безпосередній збір інформації від кінцевих користувачів. Цей рівень включає розумні датчики, актуатори та різноманітні пристрої, що фіксують дані з фізичного середовища та передають їх для подальшої обробки.

Мережевий рівень функціонує як серверно-орієнтована платформа, що забезпечує надійне з’єднання між IoT-пристроями та інфраструктурою обробки даних. Він відповідає за передачу інформації до розумних об’єктів, хмарних серверів та мережевого обладнання, гарантуючи безперебійну комунікацію між компонентами системи.

Прикладний рівень є вершиною архітектури, де формуються готові програмні рішення для кінцевих користувачів. Саме тут створюється інтерфейс взаємодії між оператором системи та клієнтом, надаючи зручний доступ до функціоналу IoT-екосистеми.

Така багаторівнева структура забезпечує комплексну функціональність системи, можливість горизонтального та вертикального масштабування, високу доступність сервісів, а також спрощує процеси технічного обслуговування та ремонтопридатності всієї IoT-архітектури.

Технологія Інтернету речей революційно змінює наше повсякденне життя, забезпечуючи суттєве підвищення комфорту та ефективності. Розумні пристрої автоматизують рутинні завдання, які раніше вимагали постійної участі людини, та виконують складні процеси, що дозволяє заощаджувати цінний час і ресурси. Однак, як стверджує відома приказка, з великою силою приходить велика відповідальність.

Поряд із численними перевагами, які надають IoT-пристрої, пропорційно зростає і спектр потенційних загроз безпеці. Ці виклики можуть становити серйозну небезпеку як для приватних користувачів, так і для організацій будь-якого масштабу – від малого бізнесу до великих корпорацій. Особливу тривогу викликають вразливості IoT-систем, інтегрованих у критично важливу інфраструктуру суспільства.

Найбільш небезпечними є ризики, пов’язані з медичними приладами, які підтримують життєво важливі функції пацієнтів, а також з промисловими системами керування на підприємствах, що становлять основу національної економіки. Кожен новий підключений пристрій розширює потенційну поверхню атаки, створюючи додаткові точки входу для зловмисників. Це збільшує

ймовірність несанкціонованого доступу до корпоративних мереж, витоку конфіденційних даних та руйнівних кібератак на критичну інфраструктуру компаній.

Сучасний стан кібербезпеки Інтернету речей характеризується значним дисбалансом між темпами технологічного прогресу та розвитком засобів захисту. Ця проблема набуває особливої актуальності в контексті експоненціального зростання кількості підключених пристроїв у глобальній мережі.

Головна причина вразливості IoT-екосистеми криється у надзвичайно динамічному розвитку самої технології. Виробники прагнуть якнайшвидше вивести свою продукцію на ринок, щоб випередити конкурентів та задовольнити зростаючий попит споживачів. У цій гонці за інноваціями питання інформаційної безпеки часто відсуваються на периферію пріоритетів розробників. Швидкість, з якою з’являються нові моделі та функціональні можливості пристроїв, значно перевищує можливості індустрії кібербезпеки адекватно реагувати на нові виклики та розробляти відповідні механізми захисту.

Виробники IoT-пристроїв зосереджують свої зусилля передусім на трьох ключових аспектах: максимальній функціональності, зручності використання та доступній ціні. Саме ці характеристики визначають конкурентоспроможність продукту на ринку та формують споживчий попит. Питання безпеки, на жаль, розглядається як вторинний фактор, що призводить до створення мільйонів потенційно вразливих точок входу в цифрову інфраструктуру.

Така ситуація формує принципово новий ландшафт кіберзагроз. Кожен новий смарт-пристрій – від побутової техніки до промислових датчиків – стає потенційним вектором атаки для зловмисників. Кіберзлочинці активно експлуатують цю уразливість, оскільки слабко захищені IoT-пристрої представляють собою легку мішень, що не вимагає значних зусиль для компрометації. Це створює замкнене коло, де масовість та доступність технології породжують нові ризики швидше, ніж індустрія встигає їх нейтралізувати.

Інтернет речей став революційною технологією, що відкрила безпрецедентні можливості для інноваційного розвитку комунікаційних систем між різноманітними об’єктами через глобальну мережу Інтернет. Ця технологія фундаментально трансформувала підходи до взаємодії пристроїв, створюючи інтелектуальну екосистему взаємопов’язаних девайсів.

Сьогодні IoT займає центральне місце в житті сучасної людини, проникаючи практично в усі сфери діяльності. Від інтелектуальних систем «розумного дому», що автоматизують побутові процеси та підвищують комфорт проживання, до складних промислових комплексів, які оптимізують виробничі процеси – технологія Інтернету речей демонструє свою універсальність та ефективність. Розумні термостати, системи безпеки, освітлення, а також промислове обладнання та логістичні системи – все це працює завдяки IoT-технологіям.

Проте експоненціальне зростання кількості підключених пристроїв призвело до виникнення серйозних викликів у сфері кібербезпеки. Масове впровадження IoT-девайсів супроводжується критичним збільшенням потенційних загроз і вразливостей, що створює ризики для конфіденційності даних, цілісності систем та безперервності їх роботи. Ця проблематика вимагає негайної та всебічної уваги від фахівців з інформаційної безпеки.

Для системного подолання цих викликів авторитетна міжнародна організація Open Web Application Security Project (OWASP) провела комплексне дослідження та сформувала перелік десяти найкритичніших вразливостей IoT-екосистеми. Цей список, відомий як OWASP IoT Top 10, базується на ретельному аналізі реальних інцидентів безпеки та представляє найбільш поширені й небезпечні загрози.

Методологія розробки цього переліку включала глибокий аналіз численних випадків компрометації IoT-систем, вивчення векторів атак та оцінку потенційних наслідків. Документ надає практичні рекомендації щодо виявлення, попередження та нейтралізації загроз, допомагаючи розробникам та

адміністраторам створювати більш захищені IoT-рішення. На рисунку 1.2 наведено топ 10 загроз та вразливостей для IoT пристроїв.

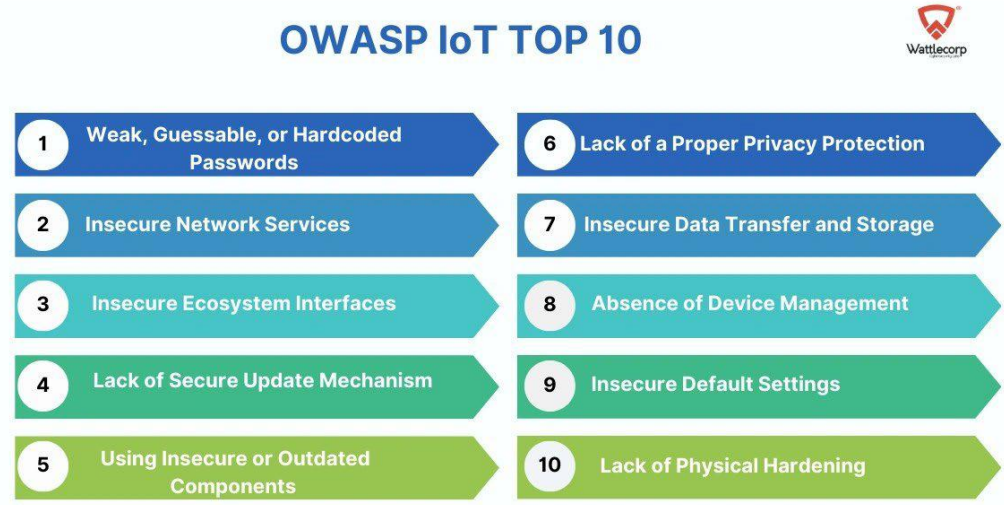


Рисунок 1.2 – OWASP Top 10 IoT [5]

Забезпечення безпеки пристроїв Інтернету речей (IoT) є одним із найважливіших і найскладніших викликів у сфері сучасної кібербезпеки. Вразливості IoT-систем мають далекосяжні наслідки, оскільки вони впливають не тільки на збереження конфіденційності та цілісності даних користувачів, але й створюють серйозні загрози для їхньої фізичної безпеки. Це перетворює IoT-пристрої на привабливі та потенційно небезпечні цілі для кіберзлочинців, які можуть використовувати їх для розгортання масштабних атак або проникнення в критичну інфраструктуру.

Для систематизації накопичених знань у галузі безпеки IoT і розробки ефективних рішень для протидії наявним загрозам було створено численні міжнародні стандарти та практичні рекомендації. Серед найвпливовіших і найвідоміших документів виділяються OWASP IoT Top 10 та ENISA Baseline Security Recommendations. Обидва стандарти відіграють критично важливу роль у формуванні підходів до безпеки IoT, однак кожен має свої унікальні особливості та сфери застосування [6].

OWASP IoT Top 10 зосереджується переважно на виявленні та аналізі найпоширеніших технічних вразливостей IoT-екосистем. Цей документ

детально описує практичні проблеми безпеки, серед яких використання слабких або стандартних паролів, відсутність належного шифрування даних при передачі та зберіганні, небезпечні налаштування пристроїв за замовчуванням, а також недостатній захист веб-інтерфейсів і мобільних додатків.

На відміну від цього, ENISA пропонує значно ширший і комплексніший підхід до забезпечення безпеки IoT. Цей стандарт інтегрує не лише технічні аспекти, але й організаційні процеси та юридичні вимоги. ENISA охоплює такі важливі напрямки, як управління повним життєвим циклом пристроїв, організація регулярних тренінгів та навчання співробітників, побудова безпечної взаємодії з третіми сторонами, а також розробка ефективних процедур управління інцидентами кібербезпеки. Завдяки цьому комплексному підходу рекомендації ENISA орієнтовані не тільки на розробників і виробників IoT-пристроїв, але й на організації різного масштабу, які інтегрують технології Інтернету речей у свої бізнес-процеси та операційну діяльність.

Європейське агентство з кібербезпеки ENISA розробило комплексні рекомендації щодо забезпечення безпеки пристроїв Інтернету речей, які наголошують на критичній необхідності системного підходу до захисту інформації. Ці настанови особливо підкреслюють важливість вбудовування механізмів безпеки на кожному етапі життєвого циклу IoT-систем – починаючи від початкового проектування та розробки, продовжуючи тестуванням і сертифікацією, та завершуючи безпосереднім впровадженням у виробниче середовище.

Ключовим аспектом рекомендацій є холістичний підхід до захисту, який передбачає врахування безпеки не лише окремого IoT-пристрою як ізольованого елемента, але й усієї взаємопов’язаної екосистеми Інтернету речей у цілому. Це охоплює захист комунікаційних каналів, серверної інфраструктури, хмарних сервісів та мобільних додатків, які взаємодіють з пристроями.

ENISA рекомендує впроваджувати багаторівневі політики безпеки, здатні ефективно протидіяти різноманітним типам загроз – від базових вразливостей до складних цілеспрямованих кібератак. Такий стратегічний підхід забезпечує

створення стійкої системи захисту, яка адаптується до динамічного ландшафту сучасних кіберзагроз та гарантує надійний захист даних користувачів протягом усього терміну експлуатації IoT-рішень.

У контексті розробки безпечних IoT-систем критично важливим аспектом є забезпечення балансу між захистом пристроїв та іншими ключовими експлуатаційними характеристиками, зокрема енергоефективністю. Цей баланс становить особливу складність для інженерів та розробників, оскільки надмірна увага до одного аспекту може негативно вплинути на інший. Зокрема, прагнення до максимальної економії енергоресурсів може призвести до компромісів у виборі криптографічних методів захисту даних. У таких випадках розробники можуть бути схильні використовувати менш ресурсомісткі, але водночас й менш надійні алгоритми шифрування, що автоматично підвищує вразливість системи перед потенційними кіберзагрозами та несанкціонованими втручаннями.

Архітектурний підхід до побудови IoT-екосистем повинен обов'язково включати принцип компартменталізації, який передбачає сегментацію системи на ізольовані компоненти. Така структурна організація створює додаткові рівні захисту, ефективно мінімізуючи потенційні наслідки як внутрішніх, так і зовнішніх атак на систему. Цей підхід цілком узгоджується з фундаментальними рекомендаціями, викладеними у документації OWASP IoT Top 10, яка наголошує на необхідності забезпечення надійного захисту програмного забезпечення та впровадження суворих механізмів контролю доступу до критично важливих системних компонентів.

Що стосується процедур верифікації безпеки, Європейське агентство з мережевої та інформаційної безпеки особливо підкреслює значущість регулярного проведення тестування на проникнення. Ці спеціалізовані тести дозволяють ідентифікувати потенційні вразливості в механізмах обробки вхідних даних, системах автентифікації та авторизації користувачів. Такий підхід безпосередньо корелює з базовими принципами OWASP IoT Top 10, які акцентують увагу на критичній важливості коректної валідації та санітизації

даних, а також на реалізації ефективних механізмів захисту від неавторизованого доступу до системних ресурсів.

Додатковим інструментом забезпечення безпеки для команд розробників є систематичне проведення аудиту програмного коду. Ретельний огляд вихідного коду дозволяє виявляти та усувати потенційні вразливості на ранніх стадіях життєвого циклу розробки, що значно знижує ризики та вартість їх подальшого усунення. Ця практика повністю відповідає філософії безпечної розробки, пропагованої OWASP, яка орієнтована на проактивне виявлення та ліквідацію проблем безпеки ще до етапу розгортання продукту в експлуатаційному середовищі.

Європейське агентство з кібербезпеки приділяє особливу увагу питанням конфіденційності в системах Інтернету речей, підкреслюючи критичну важливість інтеграції механізмів захисту приватності на всіх без винятку етапах проектування та розробки IoT-рішень.

Одним із ключових принципів, які пропагує ENISA, є концепція «Privacy by Design» – вбудовування конфіденційності за замовчуванням. Цей підхід передбачає, що захист персональних даних та приватності користувачів має бути невід’ємною складовою архітектури пристрою з самого початку розробки, а не додаватися як опціональна функція на фінальних стадіях. Практична реалізація цього принципу означає, що розробники повинні систематично ідентифікувати, аналізувати та імплементувати відповідні механізми безпеки на кожному етапі життєвого циклу продукту – від концептуального проектування до фінального впровадження та подальшої експлуатації системи.

Окрім інтеграції конфіденційності в дизайн, ENISA рекомендує обов’язкове проведення детальної оцінки впливу на конфіденційність, яка відома як Privacy Impact Assessment (PIA). Це комплексне тестування має виконуватися до офіційного запуску будь-яких нових додатків чи сервісів IoT. Головна мета такої оцінки полягає в систематичному виявленні потенційних загроз і вразливостей, які можуть негативно вплинути на приватність користувачів. Результати PIA надають розробникам цінну можливість або усунути виявлені

недоліки та вразливості на ранніх стадіях, або принаймні розробити стратегію управління ймовірними ризиками та підготувати відповідні заходи реагування.

Ці рекомендації ENISA безпосередньо корелюють з однією з найсерйозніших проблем безпеки, визначених у рейтингу OWASP IoT Top 10, а саме – відсутністю належного захисту конфіденційності (Lack of Proper Privacy Protection). Ігнорування або недостатня реалізація механізмів захисту приватності може спричинити серйозні наслідки: від витоку чутливих персональних даних і порушення фундаментального права користувачів на приватність до серйозних юридичних проблем, пов’язаних з невідповідністю законодавчим вимогам, зокрема регламенту GDPR.

Управління активами відіграє критичну роль у забезпеченні кібербезпеки сучасних організацій, що підтверджується рекомендаціями ENISA. Агентство наголошує на необхідності створення та постійної підтримки комплексних процедур управління активами разом із контролем конфігурацій для всіх критичних систем інфраструктури.

Ефективне управління активами передбачає формування чіткої та структурованої стратегії, яка охоплює повний життєвий цикл усіх технологічних компонентів. Кожен пристрій, програмне забезпечення та мережевий елемент повинні пройти процес ідентифікації, отримати відповідну класифікацію за рівнем критичності та підлягати систематичному моніторингу. Паралельно з цим необхідно здійснювати детальне документування конфігураційних параметрів кожного активу.

Така методологія приносить численні переваги для організації. По-перше, вона забезпечує оперативне виявлення потенційних загроз безпеці та вразливостей у системі. По-друге, дозволяє швидко реагувати на інциденти та мінімізувати їх наслідки. По-третє, створює основу для оптимального розподілу ресурсів та прийняття обґрунтованих управлінських рішень. Крім того, систематизований підхід до управління активами сприяє підвищенню загальної операційної ефективності та зменшенню ризиків простою критичних систем.

Таким чином, системна інтеграція принципів конфіденційності в архітектуру IoT-пристроїв є не просто рекомендацією, а критично важливою необхідністю для мінімізації ризиків, пов’язаних з недостатнім захистом приватності в сучасному взаємопов’язаному цифровому середовищі.

1.2 Огляд і аналіз методів та засобів безпеки IoT для вирішення проблеми дослідження

Сучасні великі корпорації та організації зіткнулися з безпрецедентними викликами у сфері впровадження та управління системами Інтернету речей. Масштаби цієї проблеми справді вражають: підприємствам доводиться не просто розгортати тисячі взаємопов’язаних пристроїв у межах однієї екосистеми IoT, а й керувати потенційно сотнями або навіть тисячами різноманітних кінцевих точок, кожна з яких має свої унікальні характеристики та вимоги до безпеки.

Ситуація ускладнюється тим фактом, що кожна окрема реалізація IoT може кардинально відрізнятися від іншої як за своїм призначенням, так і за функціональними можливостями. Візьмемо, к прикладу, велику роздрібну мережу: така організація може одночасно експлуатувати складні RFID-системи для точного відстеження товарних запасів у режимі реального часу, використовувати мережу бездротових маячків у торгових приміщеннях для збору та аналізу персоналізованих даних про поведінку покупців, а також активно впроваджувати передові технологічні рішення, включаючи підключені до мережі транспортні засоби для логістики, безпілотні літальні апарати для моніторингу складських приміщень та роботизовані системи для автоматизації різноманітних бізнес-процесів.

Перед фахівцями з інформаційної безпеки постає складне та багатогранне завдання: вони повинні ретельно проаналізувати, детально вивчити та чітко охарактеризувати кожну з цих розрізнених, часто технологічно несумісних систем. Більше того, необхідно розробити та впровадити відповідний життєвий

цикл безпеки, спрямований на забезпечення стабільного та захищеного функціонування всієї корпоративної інфраструктури IoT.

У цьому контексті розглядається комплексний життєвий цикл безпеки систем Інтернету речей, який органічно інтегрується в усі етапи – від первинної розробки через інтеграцію до фінального розгортання. Ключова особливість цього життєвого циклу полягає в його ітеративній природі, що дозволяє організаціям поступово, безпечно та контрольовано додавати нові можливості IoT у масштабах усього підприємства, не створюючи при цьому додаткових уразливостей.

Життєвий цикл охоплює широкий спектр тематичних напрямків, політик безпеки та операційних процедур, забезпечуючи створення динамічної системи захисту IoT-інфраструктури. Ця система постійно оновлюється, адаптується та еволюціонує відповідно до унікальних операційних потреб конкретного підприємства, враховуючи його специфіку та мінливе середовище кіберзагроз.

Сучасні організації, які виступають кінцевими споживачами технологій Інтернету речей, мають у своєму розпорядженні широкий спектр можливостей для впровадження та розгортання IoT-функціоналу. Стратегії імплементації таких систем можуть суттєво відрізнятися залежно від потреб, ресурсів та технічної спроможності конкретної компанії.

Певна частина організацій обирає шлях самостійної розробки систем Інтернету речей, створюючи власні рішення з нуля. Це дозволяє максимально адаптувати функціональність під специфічні бізнес-вимоги. Однак, такий підхід вимагає значних інвестицій у розробку, технічну експертизу та підтримку.

Водночас, ринок пропонує численні альтернативні варіанти у вигляді готових комплексних рішень. Організації можуть придбати попередньо інтегровані системи, які містять повний набір необхідних компонентів: розумні пристрої з уже інстальованим програмним забезпеченням, засоби мережевого підключення для забезпечення безперебійної комунікації, інтерфейси для взаємодії між різними мережевими елементами, а також потужні системи

аналітики для обробки великих обсягів даних та резервного копіювання інформації.

Крім того, існують гібридні моделі, що поєднують окремі елементи цих підходів, надаючи організаціям гнучкість у виборі оптимального рішення для досягнення своїх стратегічних цілей у сфері цифрової трансформації.

Платформа Splunk являє собою потужне комплексне рішення, призначене для ефективного збирання, надійного зберігання, ретельного опрацювання та глибокого аналізу машинних даних, зокрема системних логів та журналів подій. На сучасному етапі ця система демонструє надзвичайно високий рівень популярності на американському та європейському ринках, при цьому активно розширюючи свою присутність в інших географічних регіонах світу [7].

Ключовою перевагою платформи виступає її універсальність – здатність інтегруватися та працювати з даними від практично будь-яких типів пристроїв, обладнання та систем. Завдяки цій особливості сфера можливого застосування Splunk є надзвичайно різноманітною та охоплює численні галузі діяльності. Можливості інтеграції у персоналізовані дашборди продемонстровано на рисунку 1.3.

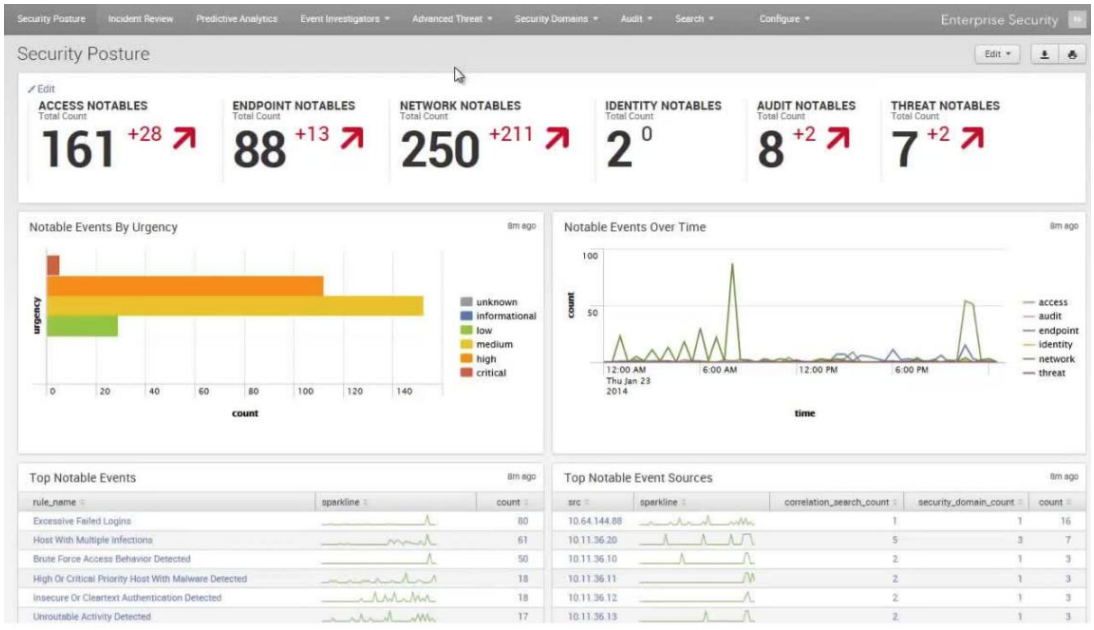


Рисунок 1.3 – Приклад візуалізації Spunk [8]

Технологічно платформа автоматично розбирає вхідні потоки інформації на окремі поля та відповідні значення для подальшої обробки. Ця обробка здійснюється за допомогою спеціалізованої мови запитів SPL (Splunk Processing Language) – унікальної розробки компанії Splunk. Використовуючи SPL, користувачі можуть створювати складні вибірки даних, формувати різноманітні таблиці, виконувати сортування та фільтрацію, проводити агрегацію інформації, генерувати детальні звіти, створювати обчислювані поля, працювати як із внутрішніми, так і зовнішніми довідниками, а також розробляти інформативні візуалізації з широким спектром графічних представлень. Додатково система підтримує налаштування автоматичних сповіщень.

До основних переваг Splunk відносять:

– потужна аналітика: Splunk забезпечує аналіз величезних обсягів даних у реальному часі, виявляючи аномалії та потенційні загрози швидко та ефективно;

– централізоване управління: платформа збирає логи з різних джерел (сервери, мережеве обладнання, додатки) в єдиному місці, спрощуючи моніторинг інфраструктури;

– розширені можливості пошуку: SPL (Search Processing Language) дозволяє створювати складні запити для глибокого аналізу подій безпеки;

– візуалізація та звітність: інтерактивні дашборди та налаштовувані звіти допомагають швидко оцінити стан безпеки;

– інтеграція: підтримує інтеграцію з численними системами безпеки та SIEM-рішеннями.

Серед недоліків варто звернути увагу на:

– висока вартість: ліцензування базується на обсязі даних, що робить Splunk дорогим рішенням для великих організацій;

– складність налаштування: потребує кваліфікованих фахівців для правильного розгортання та оптимізації;

– ресурсомісткість: вимагає потужної інфраструктури для обробки великих обсягів даних;

– крива навчання: SPL та інші функції потребують часу для опанування співробітниками;

– обмеження безкоштовної версії: free-версія має суттєві обмеження за функціоналом та обсягом даних.

Компанія McAfee [9] пропонує своїм клієнтам максимально гнучкий підхід до впровадження систем захисту інформації. Їхні рішення доступні в різноманітних форматах: як традиційні фізичні пристрої, так і сучасні віртуальні апарати, а також програмне забезпечення для встановлення на існуючу інфраструктуру. Така універсальність дозволяє організаціям обирати оптимальний варіант розгортання залежно від їхніх технічних вимог, бюджету та особливостей IT-середовища.

Архітектура продуктів McAfee побудована за модульним принципом, що забезпечує надзвичайну гнучкість у використанні. Кожен модуль може функціонувати як автономно, вирішуючи специфічні завдання безпеки, так і в комплексі з іншими компонентами, створюючи цілісну екосистему захисту. Це дає можливість компаніям поступово нарощувати свої можливості в сфері інформаційної безпеки відповідно до зростання потреб та доступних ресурсів.

Центральним елементом екосистеми безпеки McAfee виступає модуль Enterprise Security Manager (ESM). На рисунку 1.4 відображено вікно візуалізації McAfee Enterprise Security Manager.

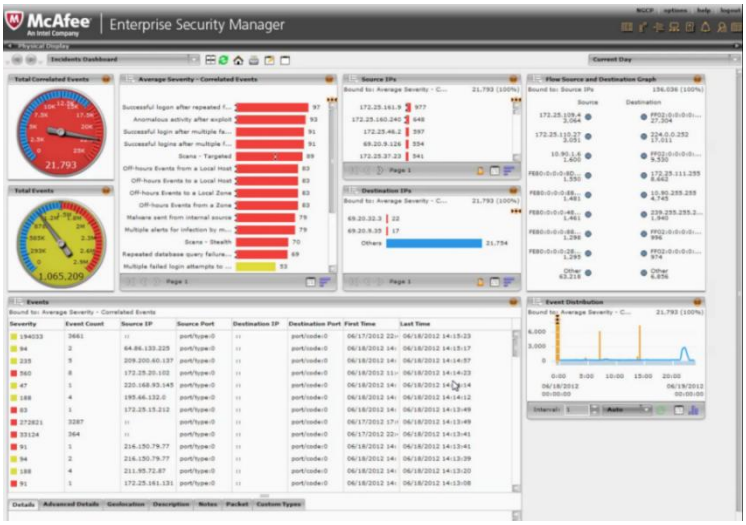


Рисунок 1.4 – Приклад візуалізації McAfee Enterprise Security Manager [10]

Цей потужний інструмент забезпечує безперервний моніторинг усієї корпоративної IT-інфраструктури в режимі реального часу. ESM систематично збирає та аналізує величезні обсяги інформації про потенційні загрози та ризики безпеці, консолідуючи дані з численних джерел. Система оснащена розумними механізмами пріоритезації, які автоматично визначають найбільш критичні загрози, що потребують негайного реагування. Завдяки цьому фахівці з безпеки можуть ефективно розподіляти свої зусилля та швидко проводити детальні розслідування інцидентів.

Додаткову цінність ESM надає інтеграція з платформою McAfee Global Threat Intelligence. Ця передова система значно розширює стандартні можливості SIEM-рішень, забезпечуючи доступ до постійно оновлюваної глобальної бази даних про кіберзагрози з усього світу. Завдяки цьому ESM може в реальному часі ідентифікувати події, пов’язані з підозрілими IP-адресами, доменами та іншими індикаторами компрометації, що суттєво підвищує ефективність захисту від сучасних кіберзагроз.

Архітектура Elastic Stack побудована на базі кількох ключових компонентів, які працюють у тісній інтеграції. До основних елементів стеку належать високопродуктивна пошукова система, що забезпечує швидкий пошук по індексованим даним, спеціалізований генератор журналів для збору та обробки логів, інтуїтивно зрозумілий веб-інтерфейс користувача для взаємодії з системою, а також розгалужена колекція відправників даних, які забезпечують надходження інформації з різних джерел. Важливою особливістю є те, що всі ці компоненти розповсюджуються під ліцензіями відкритого вихідного коду, що забезпечує прозорість, гнучкість та можливість адаптації під специфічні потреби бізнесу.

Значною віхою в розвитку екосистеми став червень 2019 року, коли компанія Elastic представила Elastic SIEM – перше офіційне рішення для управління інформацією та подіями безпеки від Elastic. Цей інноваційний продукт відкрив нові горизонти для фахівців з кібербезпеки, надавши їм потужний інструмент для комплексного аналізу подій безпеки, пов’язаних як з

хостами, так і з мережевою інфраструктурою. Elastic SIEM дозволяє проводити глибокі розслідування інцидентів безпеки через систему налаштовуваних сповіщень або за допомогою інтерактивного пошуку в режимі реального часу. Оскільки продукт з'явився на ринку відносно недавно, він ще не був включений до престижних аналітичних звітів галузі, зокрема до квадранта Gartner для SIEM-рішень. Візуалізацію подій за допомогою дашборду Kibana представлено на рисунку 1.5.

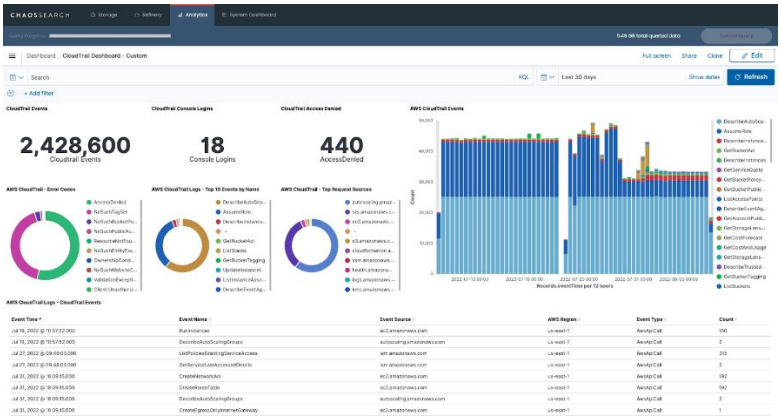


Рисунок 1.5 – Візуалізація подій за допомогою Kibana [11]

Elastic Stack виділяється серед конкурентів завдяки кільком ключовим перевагам. По-перше, продукти характеризуються високою швидкодією та оптимізованою продуктивністю, що особливо важливо при обробці великих масивів даних. По-друге, вони не вимагають надпотужних апаратних ресурсів, що робить їх доступними для організацій з різними бюджетами. По-третє, розробники приділили значну увагу створенню логічного та інтуїтивно зрозумілого інтерфейсу, що дозволяє звичайним користувачам без глибоких технічних знань ефективно працювати з системою та швидко освоювати її функціонал.

Програмний пакет Elastic Stack [12] пропонує гнучку модель ліцензування, яка включає безкоштовне використання базового функціоналу для локального розгортання. Водночас компанія надає розширені комерційні плани з професійною технічною підтримкою для корпоративних клієнтів. SIEM-система

від Elastic інтегрована як окремий модуль до платформи візуалізації даних Kibana, що забезпечує комплексний моніторинг безпеки.

Усі компоненти екосистеми Elastic доступні у форматі хмарних SaaS-рішень, проте варто зазначити, що хмарні версії не мають безкоштовного тарифного плану. Що стосується окремих продуктів стеку, то Logstash, Beats та Kibana розповсюджуються з відкритим вихідним кодом і є повністю безкоштовними для використання. Натомість Elasticsearch вимагає придбання ліцензії для комерційного застосування.

Важливою перевагою Elastic Stack є висока сумісність і модульність архітектури. Кожен компонент стеку можна інтегрувати з численними сторонніми інструментами та рішеннями, що створює гнучке середовище для обробки даних. Особливо популярною серед спеціалістів є платформа Kibana, яка завдяки потужним можливостям візуалізації та аналітики широко використовується не лише в межах екосистеми Elastic, але й у комбінації з іншими системами моніторингу та аналізу даних.

1.3 Постановка завдання на кваліфікаційну роботу магістра

Сучасний розвиток Інтернету речей (Internet of Things, IoT) призводить до зростання кількості підключених пристроїв, які генерують великі обсяги даних у реальному часі. Однак розширення IoT-інфраструктури супроводжується збільшенням кількості вразливостей, що створює потенційні ризики несанкціонованого доступу, втрати або модифікації даних. Тому важливим аспектом є забезпечення належного рівня інформаційної безпеки IoT-систем шляхом моніторингу, аналізу та своєчасного реагування на загрози.

Метою даної кваліфікаційної роботи є дослідження та розробка підходу до аналізу безпеки IoT-архітектури за допомогою інструментарію ELK Stack (Elasticsearch, Logstash, Kibana). Використання ELK Stack дозволяє централізовано збирати журнали подій із пристроїв IoT, виконувати їх

фільтрацію, зберігання, кореляцію та візуалізацію для виявлення потенційних аномалій і загроз.

Основні завдання дослідження, що передбачають досягнення поставленої мети, включають в себе проведення аналізу сучасних загроз безпеці IoT-систем та методів їх виявлення; дослідження архітектурних особливостей побудови систем безпеки IoT з використанням компонентів ELK Stack; розробки структуру модуля безпеки з визначенням ключових компонентів збору, агрегації та аналізу логів; імплементацію механізму виявлення аномалій та підозрілої активності на основі машинного навчання; розробку системи візуалізації безпекових подій та dashboard для моніторингу в Kibana; проведення тестування розробленого модуля на тестовому стенді IoT-інфраструктури; виконання порівняльного аналізу ефективності запропонованого рішення з існуючими аналогами. Очікуваним результатом роботи є створення прототипу модуля моніторингу безпеки для IoT-систем, який забезпечить виявлення загроз у режимі реального часу та сприятиме підвищенню рівня інформаційної безпеки в розподілених мережах Інтернету речей.

Висновки до розділу 1

Аналіз проблематики захисту IoT-систем продемонстрував, що безпека інформації в архітектурі IoT залишається критичним викликом через її розподілений характер, різноманітність пристроїв та обмеженість ресурсів. Модуль безпеки інформації вимагає комплексного підходу, що охоплює аутентифікацію, авторизацію, шифрування та моніторинг.

Дослідження показало, що ELK Stack є високоефективним рішенням для агрегації, аналізу та візуалізації логів і подій безпеки в масштабі IoT. Цей стек надає необхідні інструменти для виявлення аномалій та загроз у реальному часі, що є ключовим для проактивного захисту. Його використання значно підвищує операційну видимість стану безпеки архітектури.

РОЗДІЛ 2
ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ МОДУЛЮ БЕЗПЕКИ ІНФОРМАЦІЇ

2.1 Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання

Практична частина дослідження буде реалізована з використанням потужної платформи Elastic Stack, яка демонструє виняткову ефективність у сфері моніторингу інфраструктури, глибокого аналізу системних журналів, перевірки конфігураційних файлів та відстеження критичних подій інформаційної безпеки. Збір та передача даних здійснюватиметься через спеціалізовані агенти сімейства Beats, зокрема Filebeat, що забезпечить надійну доставку логів до центральної системи обробки. Архітектура експериментального середовища передбачає розподілену топологію з декількох незалежних хостів, кожен з яких виконуватиме специфічну роль та міститиме відповідний компонент екосистеми Elastic Stack, що дозволить створити масштабовану та відмовостійку систему моніторингу безпеки.

Апаратна складова інфраструктури включає широкий спектр обладнання: потужні серверні системи різної конфігурації, високопродуктивні мережеві комутатори для забезпечення швидкої передачі даних, інтелектуальні маршрутизатори для керування трафіком, а також додаткові спеціалізовані пристрої. Ці елементи забезпечують необхідні обчислювальні потужності, достатню пропускну здатність каналів зв’язку та стабільні комунікаційні ресурси для функціонування віртуальної інфраструктури.

Програмна компонента охоплює сучасні віртуалізаційні платформи, ефективні гіпервізори різних типів, спеціалізовані інструменти для централізованого управління віртуальними мережами, системи моніторингу та автоматизації, а також інноваційні рішення, орієнтовані на максимальну оптимізацію продуктивності, безпеки та масштабованості віртуального середовища.

Процес налагодження та створення віртуальних мереж і середовищ передбачає застосування комплексного набору апаратних і програмних засобів, детальна специфікація яких представлена в таблиці 2.1. Слід наголосити, що обґрунтований та виважений підбір цих компонентів становить фундаментальний етап у побудові продуктивного, надійного та оптимізованого віртуального середовища, здатного відповідати сучасним вимогам інформаційних технологій.

Таблиця 2.1 – Апаратні та програмні компоненти

	Операційна система	Відкриті порти	Реалізовані сервіси
Сервер	Ubuntu server 24.04.3 LTS	Elasticsearch: 9200 TCP Kibana: 5601 TCP Logstash: 5044 TCP Filebeat: 5044 TCP	ELK Stack 8 Filebeat
Кінцеві точки	Ubuntu 24.04 Windows 11	Filebeat: 5044 TCP	Filebeat

Ключовим елементом успішної реалізації віртуальної інфраструктури є грамотна конфігурація та злагоджена взаємодія всіх її складових компонентів. Правильне налаштування системи гарантує максимальну продуктивність, оптимальну ефективність роботи мережі та стабільність функціонування. Комплексний підхід до інтеграції компонентів дає змогу сформувати надійне, гнучке та масштабоване середовище, здатне адаптуватися до змінних вимог. Така архітектура повністю відповідає технічним потребам проекту, забезпечує безперебійну роботу сервісів та гарантує високий рівень задоволеності кінцевих користувачів системою.

Для забезпечення оптимальної функціональності системи було здійснено детальне налаштування операційної системи та мережевих портів, що є критично важливим для злагодженої взаємодії між різними програмними компонентами. Серверна інфраструктура, розгорнута на платформі Ubuntu 24.04, використовує чітко визначені TCP-порти для кожного сервісу: Elasticsearch функціонує через порт 9200, Kibana забезпечує візуалізацію даних через порт 5601, а Logstash та

Filebeat взаємодіють через порт 5044. Така конфігурація мережевих з'єднань гарантує безперебійну комунікацію, надійний обмін логами та ефективну синхронізацію даних між усіма елементами екосистеми ELK Stack версії 8 та агентом збору Filebeat.

2.2 Практична реалізація об'єкта проектування

Практична частина дослідження здійснювалась на робочій станції DELL, оснащеній восьмиядерним процесором Intel(R) Xeon(R) CPU E5-2650 v2 з тактовою частотою 2.60GHz та оперативною пам'яттю обсягом 16 ГБ. Цей апаратний комплекс використовувався як фізичний хост для розгортання віртуалізованого середовища розробки та тестування системи. Архітектура експериментального стенду включає три віртуальні машини, які функціонують паралельно, здійснюють активну взаємодію між собою через мережеві інтерфейси та слугують основними об'єктами для проведення імітаційного моделювання та валідації розробленої системи.

Процес розгортання операційної системи розпочався з ініціалізації віртуальної машини та подальшого встановлення Ubuntu server 24.04.3 LTS. На початкових етапах інсталяції було виконано налаштування локалізації, що включало визначення мови інтерфейсу, конфігурацію розкладки клавіатури відповідно до регіональних стандартів та встановлення часового поясу. Серед доступних варіантів встановлення обрано опцію «Normal Installation», яка забезпечує комплексну інсталяцію повного пакету офісних додатків та стандартних утиліт системи. Наступним кроком стало створення облікового запису користувача з реєстрацією персональних даних та встановленням надійного пароля для захисту системи. Після успішного завершення всіх етапів встановлення система автоматично ініціювала перезавантаження для коректного застосування конфігурацій.

Для оптимізації взаємодії між хост-системою та віртуальним середовищем було виконано монтування образу VBoxGuestAdditions.iso через віртуальний

оптичний привід. У налаштуваннях віртуальної машини активовано функцію двонапрямленого обміну даними, що дозволяє використовувати спільний буфер обміну між операційними системами.

Перед початком інсталяції програмного середовища Java на операційну систему було здійснено попереднє встановлення спеціалізованого пакету apt-transport-https. Цей компонент забезпечує можливість безпечного отримання доступу до програмних репозиторіїв через захищений протокол передачі даних HTTPS. Надалі було проведено процедуру встановлення платформи OpenJDK версії 11 як на операційній системі Ubuntu, так і на Windows. Після завершення інсталяції виконано верифікацію встановленої версії Java для підтвердження коректності налаштування робочого середовища розробки. На рисунку 2.1 проілюстровано вікно встановлення пакету доступу сховища через HTTPS.

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu4).
ca-certificates is already the newest version (20240203~22.04.1).
curl is already the newest version (7.81.0-1ubuntu1.21).
gnupg is already the newest version (2.2.27-3ubuntu2.4).
The following packages were automatically installed and are no longer required:
  bridge-utils libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib python-pkg-resources
  python2-minimal python2.7 python2.7-minimal python3-distlib python3-filelock python3-platform
  ubuntu-fan
```

Рисунок 2.1 – Встановлення пакету доступу сховища через HTTPS

Щоб налаштувати змінну середовища для коректної роботи Java, необхідно виконати кілька послідовних кроків. Спочатку відкриваємо конфігураційний файл «/etc/environment» за допомогою текстового редактора nano з правами адміністратора. У цьому файлі додаємо рядок «JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64», який вказує системі шлях до встановленої версії Java Development Kit. Це критично важливо для забезпечення стабільної роботи Java-додатків та інструментів розробки. Після збереження змін у файлі виконуємо команду «source/etc/environment», яка негайно завантажує оновлені змінні середовища без необхідності перезавантаження системи. Завершальним етапом є верифікація правильності

налаштування шляхом перевірки значення змінної. Після успішного виконання цих операцій Java-середовище повністю сконфігуроване та готове для ефективної роботи з проєктами.

Початковий етап передбачає встановлення Elasticsearch у систему з використанням службових команд, що наведено на лістингу 2.1.

Лістинг 2.1 – Конфігурація Elasticsearch

```
sudo apt install elasticsearch
sudo systemctl start elasticsearch
sudo systemctl enable elasticsearch
```

кінець лістингу 2.1

Наступним кроком завантажуємо офіційний відкритий ключ підпису для забезпечення безпеки та автентичності пакета. Далі виконуємо безпосереднє встановлення Elasticsearch, використовуючи спеціалізовані команди, які детально показані на відповідному рисунку, після чого ініціюємо запуск системи. Завершальним кроком є верифікація статусу роботи Elasticsearch через перевірку його активності, що дозволяє переконатися у коректному функціонуванні системи та готовності до подальшої роботи. Активація та розгортання elasticsearch наведено на рисунку 2.2

```
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2025-12-01 18:03:34 EET; 40s ago
       Docs: https://www.elastic.co
   Main PID: 1174 (java)
      Tasks: 150 (limit: 9428)
     Memory: 3.8G
        CPU: 2min 12.579s
     CGroup: /system.slice/elasticsearch.service
             ├─1174 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.na
             ├─2664 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.ne
             └─2720 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controlle

грy 01 18:02:40 ELK systemd[1]: Starting Elasticsearch...
грy 01 18:03:34 ELK systemd[1]: Started Elasticsearch.
```

Рисунок 2.2 – Вікно розгортання elasticsearch

Процес конфігурації Elasticsearch здійснюється шляхом редагування конфігураційного файлу «/etc/elasticsearch/elasticsearch.yml» за допомогою

текстового редактора nano. На першому етапі налаштування необхідно перейти до розділу «Network», де потрібно видалити символ коментаря біля параметра network.host та встановити його значення як network.host: 0.0.0.0, що дозволяє прослуховувати всі мережеві інтерфейси замість локального IP-адреси системи. Крім того, у розділі «Discovery» слід додати новий рядок з параметром discovery.seed_hosts: [ ], що визначає список вузлів для виявлення кластера. Другий етап конфігурації Elasticsearch передбачає модифікацію налаштувань автоматичної конфігурації безпеки на початку файлу, де необхідно змінити значення параметра з true на false. Після завершення всіх модифікацій конфігураційного файлу обов'язковим є перезапуск сервісу Elasticsearch командою sudo systemctl restart elasticsearch для застосування внесених змін.

Для верифікації коректної роботи Elasticsearch виконаємо перевірку його функціональності та доступності сервісу через веб-браузер, перейшовши за адресою http://10.0.2.15:9200. Отримані позитивні результати тестування дозволяють нам впевнено рухатися далі та переходити до наступних етапів конфігурації системи, включаючи детальне налаштування індексів, mapping'ів та інших компонентів пошукової інфраструктури. На рисунку 2.3 продемонстровано вікно перевірки функціоналу elasticsearch.

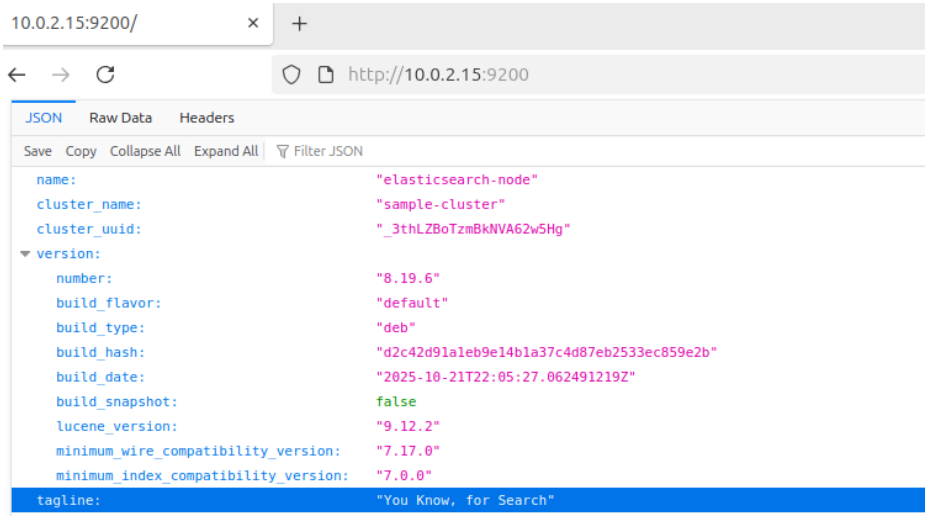


Рисунок 2.3 – Перевірка роботи Elasticsearch в браузері

Тепер переходимо до наступного важливого етапу – встановлення та налаштування Logstash. Цей потужний інструмент відіграє ключову роль у нашій інфраструктурі, оскільки він відповідає за збір, обробку та передачу даних з різноманітних джерел. Зібрана інформація надалі аналізується та візуалізується у Kibana, а також зберігається у базі даних Elasticsearch для подальшого використання.

Процес встановлення Logstash виконується за тим самим принципом, який ми застосовували раніше під час роботи з Elasticsearch. Послідовно виконуємо встановлення пакету, запускаємо службу та додаємо її до автозавантаження системи. Після завершення цих кроків обов'язково перевіряємо статус служби командою sudo systemctl status logstash, щоб переконатися у коректності роботи. Отриманий результат, що відображено на рисунку 2.4, підтверджує успішну активацію та належне функціонування Logstash. На цьому етапі конфігураційний файл залишаємо без змін.

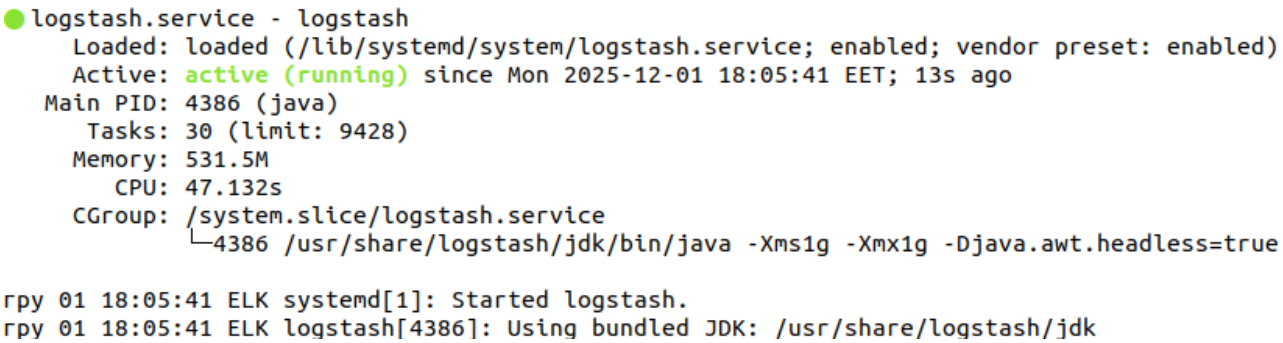


Рисунок 2.4 – Активація та запуск налаштування Logstash

Ключовим компонентом системи моніторингу виступає Kibana – потужна платформа з інтуїтивним графічним інтерфейсом, призначена для детального аналізу, візуалізації та інтерпретації накопичених логів і журнальних файлів. Цей багатофункціональний інструмент надає широкі можливості для гнучкого налаштування параметрів, комплексної конфігурації компонентів та безперешкодної інтеграції з різноманітними системами, агентами збору даних та програмним забезпеченням стороннього виробництва. Процес розгортання здійснюється за перевіреною методологією: спочатку виконується інсталяція

пакету, потім ініціюється запуск сервісу з подальшою активацією служби в системному менеджері.

На завершальному етапі проводиться верифікація операційного статусу служби, результати якої представлено на рисунку 2.5, підтверджують, що Kibana успішно функціонує в активному режимі та готова до продуктивної роботи.

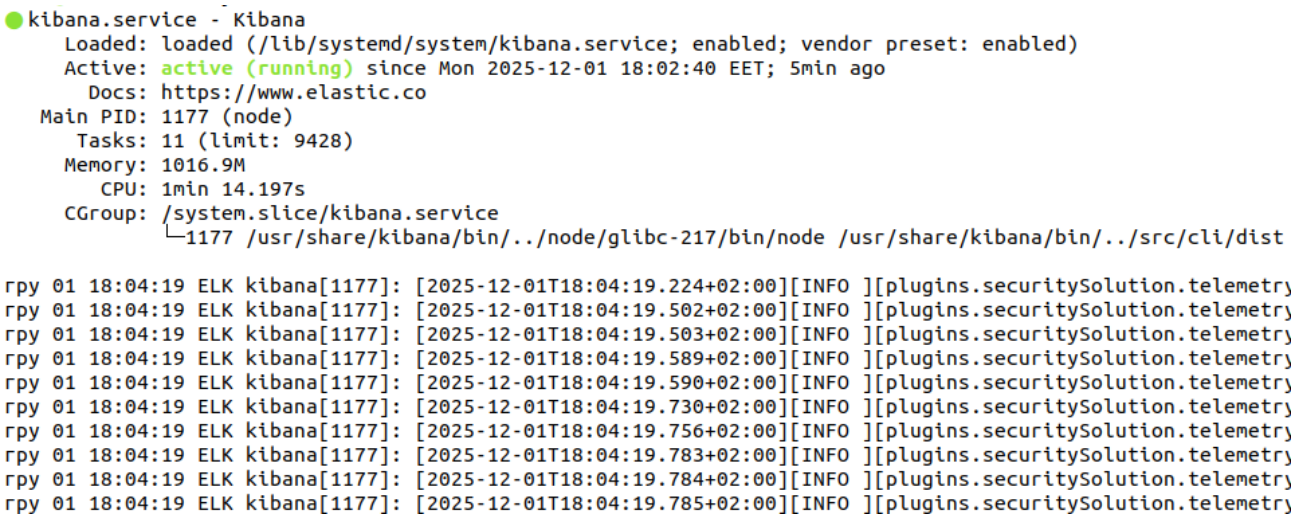


Рисунок 2.5 – Верифікація служби Kibana

Процес налаштування Kibana здійснюється через модифікацію конфігураційного файлу, розташованого за шляхом «/etc/kibana/kibana.yml», який відкривається за допомогою текстового редактора nano. У файлі визначаються ключові параметри: server.port встановлюється на значення 5601, server.host спочатку має значення «localhost», а elasticsearch.hosts вказує на [«http://localhost:9200»]. Критичним кроком є зміна параметра server.host з «localhost» на «0.0.0.0», що дозволяє сервісу прослуховувати запити з будь-якої доступної IP-адреси в мережі. Після збереження всіх внесених модифікацій обов’язково виконується перезапуск служби Kibana для застосування нової конфігурації.

Для перевірки коректної роботи та функціональних можливостей платформи Kibana необхідно виконати наступні кроки. Спочатку запустимо будь-який сучасний веб-браузер на робочій станції. Далі в адресному рядку браузера введемо та перейдемо за URL-адресою: http://10.0.2.15:5601, яка веде до

інтерфейсу Kibana. Після завантаження сторінки перед нами відкриється повнофункціональне робоче середовище Elastic Stack. Саме через цей веб-інтерфейс здійснюватиметься централізований моніторинг усіх системних подій, змін конфігурацій, активностей користувачів та дій встановленого агента. Kibana надає зручні інструменти для візуалізації даних, створення дашбордів та аналізу логів у реальному часі. На рисунку 2.6 проілюстровано вікно Elastic Stack.

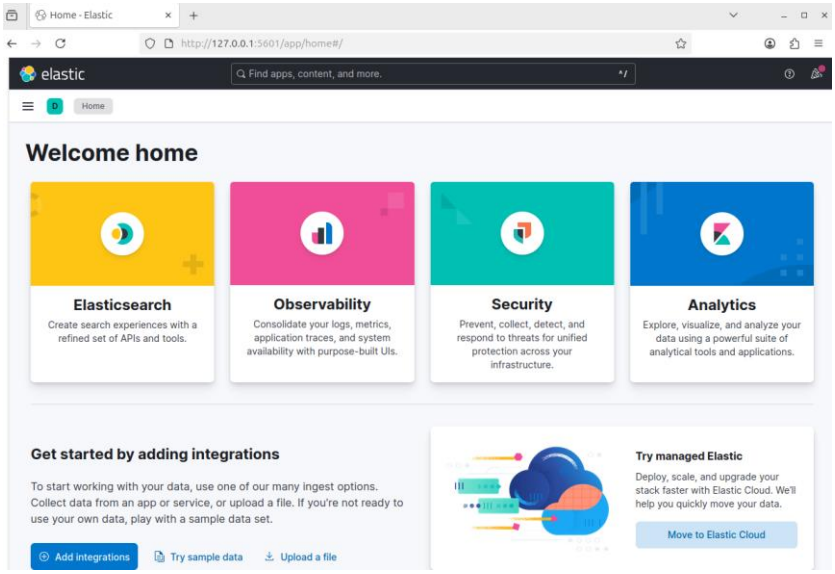


Рисунок 2.6 – Середовище Elastic

Filebeat являє собою компактний і ефективний агент збору даних, який виконує функції передачі файлів журналювання до централізованої системи обробки. Серед усієї лінійки агентів сімейства Elastic Beats саме Filebeat користується найбільшою популярністю завдяки своїй універсальності та надійності. Однією з найважливіших функціональних особливостей даного агента є інтелектуальна система контролю навантаження, яка дозволяє автоматично регулювати швидкість передачі даних у випадках, коли служба Logstash перевантажена вхідними потоками інформації. Процес встановлення агента на цільову операційну систему виконується шляхом використання спеціальної команди.

За стандартними налаштуваннями Filebeat здійснює передачу зібраних даних безпосередньо до Elasticsearch. Однак архітектура агента передбачає

гнучкість конфігурації, що дозволяє перенаправити інформацію про події безпеки (Sec_Event_Logs) до Logstash для додаткової обробки та фільтрації даних перед їх збереженням.

Щоб належним чином налаштувати систему збору логів, потрібно відредагувати конфігураційний файл Filebeat, який знаходиться в директорії /etc/filebeat/filebeat.yml. У процесі редагування необхідно деактивувати секцію «Elasticsearch Output», додавши символ коментаря (#) перед такими рядками: # output.elasticsearch, # Array of hosts to connect to та # hosts: [«localhost:9200»]. Водночас слід активувати секцію «Logstash Output», видаливши символи коментарів перед рядками output.logstash та hosts: [«localhost:5044»], що забезпечить перенаправлення даних до сервера Logstash для подальшої обробки та аналізу.

Після завершення внесення необхідних конфігураційних параметрів зберігаємо модифікований файл налаштувань та виконуємо активацію відповідного системного модуля Filebeat у нашому середовищі. На наступному етапі завантажуємо та встановлюємо необхідний шаблон індексу, який забезпечить коректну та безперебійну функціональність агента Filebeat під час передачі даних. Реалізація всіх перелічених етапів формує надійну та ефективну систему моніторингу IoT пристроїв. Для підтвердження коректності функціонування розробленого рішення необхідно здійснити тестування його компонентів. На рисунку 2.7 продемонстровано агент Filebeat в терміналі.

```
health status index                                                                uuid                   pri rep docs.count docs.deleted store.size pri.store.size dataset.size
green  open   .internal.alerts-transform.health.alerts-default-000001              HQF41gaeScG9tOSEJ-6f4w   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-observability.logs.alerts-default-000001            aUrYfouZQki-sE7MDegxiw   1   0          0            0       249b           249b         249b
yellow open   filebeat-2025.11.08                                                  Tz0dKlGDRTmdi-jSD4hnag   1   1          0            0       249b           249b         249b
green  open   .internal.alerts-observability.uptime.alerts-default-000001         uUHZ1LXnS1uok95XNhLUZw   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-ml.anomaly-detection.alerts-default-000001         t60UeCIMQRCBLFZnX2dByA   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-observability.slo.alerts-default-000001            GBllnG1bRtmNgM1btwJntg   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-observability.apm.alerts-default-000001            ygcJ0wS-SxK3Wge89DGTAg   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-default.alerts-default-000001                      RmqwLcx1SC2FGG6UF3oHYg   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-streams.alerts-default-000001                      42EgsPnAScmv4wzyGfjYAA   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-security.attack.discovery.alerts-default-000001    VM0-5j0RQ32oeYrk8TZlzw   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-observability.metrics.alerts-default-000001       AUZAGDb5SceYwSqtvPyBSA   1   0          0            0       249b           249b         249b
yellow open   .ds-cowrie-2025-2025.11.08-000001                                    Iz_2KMGUTfKixCtSGheKkA   1   1          0            0       249b           249b         249b
green  open   .internal.alerts-observability.threshold.alerts-default-000001      qyNmo2mrTtOQ9ORFAzTxyQ   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-ml.anomaly-detection-health.alerts-default-000001 8kn1XN32RyizXvot3zTMeg   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-security.alerts-default-000001                     N6FAfqOCRr-vML3HwTPX4w   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-dataset.quality.alerts-default-000001              Kj9pObRPSreNyJsyRrSO4g   1   0          0            0       249b           249b         249b
yellow open   .ds-filebeat-8.19.6-2025.11.06-000001                              8A5hdaQmQjSBMeLeLv94Ew   1   1     277445            0    689.5mb        689.5mb      689.5mb
green  open   .internal.alerts-stack.alerts-default-000001                        OyxZvfPXQJ29FQZQ83SO7A   1   0          0            0       249b           249b         249b
```

Рисунок 2.7 – Робота агента Filebeat в терміналі

Для верифікації поточного операційного стану сервісу та перевірки його працездатності використовуємо вже добре знайому системну команду sudo systemctl status filebeat. Роботу агента Filebeat з використанням сервісу elasticsearch через порт 9200 наведено на рисунку 2.8.

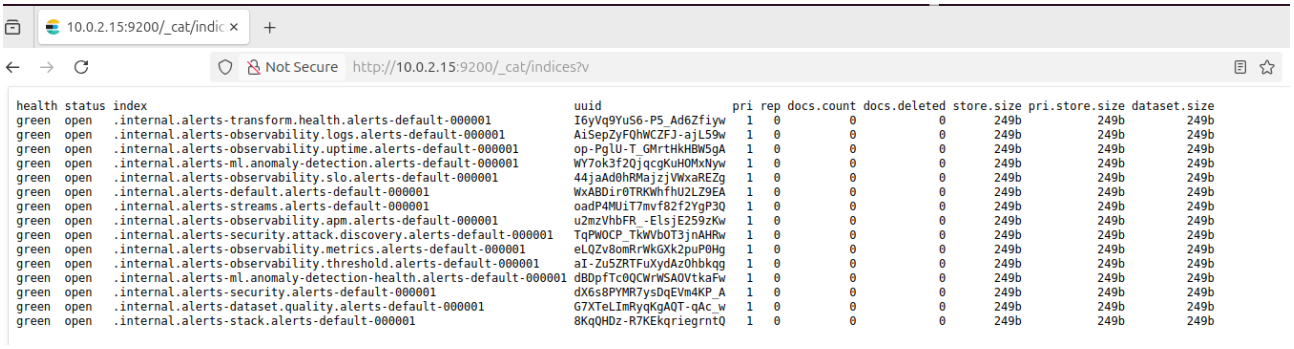


Рисунок 2.8 – Робота агента Filebeat через elasticsearch

Класифікація загроз інформаційній безпеці може здійснюватися за різними критеріями, зокрема за типом атак, які вони провокують. Для ефективного моніторингу мережевої активності та своєчасного виявлення аномального трафіку, що свідчить про потенційні загрози та існуючі вразливості системи, планується проведення контрольованої симуляції найпоширеніших типів кібератак. До переліку модельованих атак входять: фішингові атаки, спрямовані на отримання конфіденційних даних через соціальну інженерію; атаки типу man-in-the-middle, що дозволяють зловмисникам перехоплювати та модифікувати дані під час передачі; а також bruteforce-атаки, засновані на методичному переборі паролів для несанкціонованого доступу до захищених ресурсів. Така комплексна симуляція дозволить виробити ефективні механізми захисту.

Сценарій фішинг атаки будується на тому, що зловмисник надсилає email з підробленим відправником (наприклад, bank@secure-login.com замість bank.com), що містить посилання на фальшивий сайт для крадіжки облікових даних. Тестування та моніторинг через ELK Stack характеризується тим, що Logstash збирає логи з: email-серверів (відправник, тема, вкладення); веб-серверів (переходи за підозрілими URL); firewall та proxy (з’єднання з

незвичними доменами). Сервіс Elasticsearch індексує та аналізує: незвичні домени з помилками в написанні; високу кількість переходів за коротким проміжок часу; геолокацію відправників. За допомогою Kibana здійснюємо візуалізацію: дашбордів з підозрілими email-кампаніями; графіки спроб входу після фішингових листів та алерти при виявленні шаблонів атак. До основних індикаторів при атаці типу фішинг відносять невідповідність SPF/DKIM записів; підозрілі URL-шорткати; аномальна активність користувачів після отримання листа. На рисунку 2.9 відображено активності впродовж фішинг атаки.

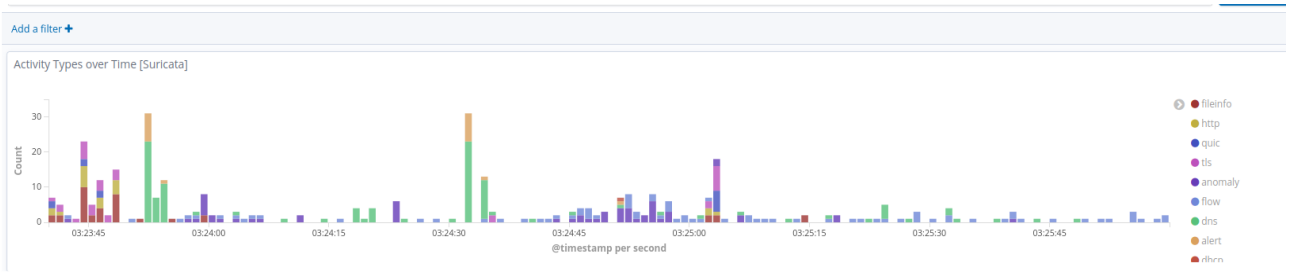


Рисунок 2.9 – Графік активності впродовж фішинг атаки

Для проведення практичної демонстрації атаки типу «Людина посередині» (Man-in-the-Middle), в контрольованому лабораторному середовищі буде використовуватись конфігурація з двох віртуальних машин, обидві з яких працюють під управлінням операційної системи Ubuntu. Атакуюча машина виконуватиме роль перехоплювача мережевого трафіку, тоді як цільова система слугуватиме об’єктом дослідження. Для встановлення з’єднання між цими віртуальними машинами планується використовувати FTP-сервер vsftpd (Very Secure FTP Daemon), який є одним з найпопулярніших та надійних FTP-серверів для Linux-систем. Такий підхід дозволить відтворити реальний сценарій перехоплення даних аутентифікації та передачі файлів у незахищеному мережевому середовищі, продемонструвавши вразливості незашифрованих протоколів передачі даних.

Під час симуляції атаки зловмисник перехоплює трафік між клієнтом і веб-сервером, використовуючи ARP spoofing для переспрямування пакетів через свою машину. Атакуючий підміняє SSL-сертифікат, встановлюючи власний

проксі-сервер між жертвою та легітимним сервісом. Моніторинг та тестування через ELK Stack характеризується тим, що Logstash збирає мережеві логи (NetFlow, Zeek); системні логи про ARP-таблиці; SSL/TLS handshake логи та DNS запити. Поряд з тим elasticsearch індексує наступні аномалії: дублювання MAC-адрес для однієї IP; незвичайні SSL-сертифікати (самопідписані, невідомі CA); раптові зміни в ARP-таблицях; підозрілі DNS відповіді. Kibana Dashboard візуалізує: спайки ARP-запитів; графік невалідних SSL-сертифікатів; географічні аномалії IP-адрес; timeline подій для кореляції. На рисунку 2.10 проілюстровано графік активності під час MITM атаки.

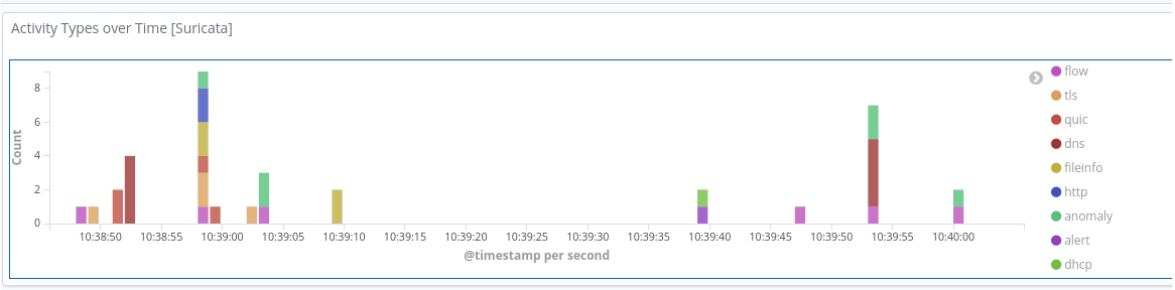


Рисунок 2.10 – Графік активності під час MITM атаки

Наступний етап практичної реалізації передбачає застосування ELK Stack як потужної платформи для аналізу логів у реальному часі та виявлення кібератак типу Brute Force. Brute Force атака полягає у систематичному переборі облікових даних (імен користувачів та паролів) з метою отримання несанкціонованого доступу до системи (наприклад, через SSH або RDP). Процес тестування включає імітацію Brute Force атаки на цільову систему. Логи автентифікації з цільової системи збираються (наприклад, за допомогою Filebeat), передаються до Logstash для парсингу та нормалізації, і зберігаються в базі Elasticsearch.

Сценарій симуляції атаки Brute Force передбачає наступні кроки:

– на сервері, що надає доступ до сервісу (наприклад, SSH або веб-додаток із формою входу), симулюється brute force-атака за допомогою інструменту

Hydra. Виконується серія швидких спроб входу з однієї IP-адреси, використовуючи список імен користувачів і паролів;

– логи автентифікації (наприклад, /var/log/auth.log у випадку SSH чи веб-серверні access/error logs) перенаправляються до Logstash, де відбувається попередня обробка: парсинг, нормалізація та маркування подій;

– у elasticsearch дані індексуються з часовими мітками та полями (IP, користувач, статус входу);

– у Kibana створюються дашборди для моніторингу підозрілих патернів: аномально висока кількість невдалих входів з одного джерела протягом короткого часу;

– на основі правил кореляції (наприклад, більше 10 невдалих спроб за 1 хвилину) генерується алерт.

Тест дозволяє перевірити ефективність збору логів, швидкість виявлення brute force-атак і правильність налаштованих порогів сповіщень. Це допомагає організації своєчасно реагувати на подібні загрози в реальних умовах. На рисунку 2.11 наведено графік активності під час BruteForce.

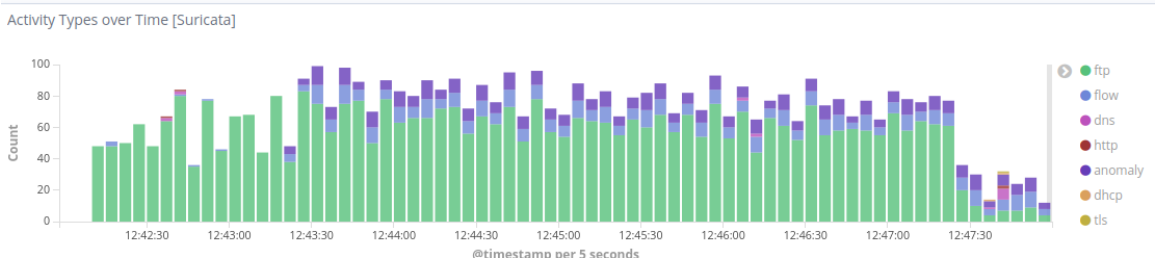


Рисунок 2.11 – Графік активності під час BruteForce

Для забезпечення комплексного моніторингу та тестування інформаційної безпеки було розроблено та впроваджено систему на основі ELK-стеку, який включає три ключові компоненти: Elasticsearch, Logstash та Kibana. Це популярне open-source рішення забезпечує ефективний захист інформаційних ресурсів організації. В процесі дослідження детально проаналізовано функціональні можливості кожного компонента стеку, визначено оптимальні параметри їхньої конфігурації для синхронізованої взаємодії.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи в рамках теоретичного дослідження та практичної реалізації було детально проаналізовано та описано комплексний процес розгортання та налаштування системи моніторингу безпеки, орієнтованої на виявлення аномалій у процесах автентифікації користувачів. Для реалізації цього рішення використовувався стек потужних інструментів з відкритим вихідним кодом, а саме: Suricata для аналізу мережевого трафіку, Elasticsearch для зберігання та індексації даних, Kibana для візуалізації та аналітики, а також Filebeat для збору та передачі логів.

Представлена методика включає покрокові інструкції з інсталяції, конфігурації та інтеграції всіх згаданих компонентів, що дозволяє побудувати повнофункціональну систему безпеки. Розроблене рішення демонструє високу ефективність у детектуванні широкого спектру кібератак, включаючи фішингові кампанії, атаки типу man-in-the-middle та спроби підбору паролів методом brute-force. Система здійснює безперервний аналіз мережевого трафіку, виявляє підозрілі патерни поведінки та ідентифікує аномальні активності, забезпечуючи своєчасне реагування на потенційні загрози інформаційній безпеці організації.

Розроблена та імплементована система моніторингу довела свою практичну життєздатність як надійне рішення для безперервного спостереження та автоматизованого виявлення аномалій в реальних виробничих умовах. Це підтверджується результатами успішно проведеної емуляції різноманітних типів кібератак та адекватним системним реагуванням на них. Використання програмних продуктів з відкритим вихідним кодом забезпечує значну технологічну гнучкість, масштабованість архітектури та широкі можливості для подальшого розширення функціоналу. Крім того, відкрита природа застосованих рішень полегшує безшовну інтеграцію з іншими корпоративними системами кібербезпеки та захисту інформації.

РОЗДІЛ 3
ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МОДУЛЮ БЕЗПЕКИ ІНФОРМАЦІЇ

3.1 Методика проведення дослідження

Інтернет речей трансформує автоматизацію у побутовій, промисловій та інфраструктурній сферах, формуючи розумні системи керування. Проте експансія підключених девайсів породжує значні загрози безпеці та приватності. Специфіка IoT-середовищ характеризується різноманітністю пристроїв та їхніми обмеженими обчислювальними можливостями. Більшість девайсів володіють недостатніми ресурсами пам’яті й продуктивності для використання класичних криптографічних технологій. Слабкий програмний захист уможливлює створення IoT-ботнетів, які провокують руйнування інфраструктури, фінансові збитки та репутаційні кризи. Парадоксально, але посилення заходів безпеки може знижувати адаптивність системи та провокувати нові вразливості.

За таких обставин перспективними стають технології машинного навчання, здатні обробляти інформаційні потоки в режимі реального часу з мінімальним споживанням ресурсів. Оскільки ручна обробка масивів даних економічно нераціональна, рішення на базі штучного інтелекту виявляються найефективнішими. Штучний інтелект являє собою комплекс технологій, що дозволяють технічним системам імітувати когнітивні здібності людини – аналіз, синтез та ухвалення рішень. Машинне навчання становить спеціалізований сегмент AI, зосереджений на автоматичному виявленні, структуруванні, опрацюванні та класифікації великих інформаційних масивів. Під час аналізу система розпізнає поведінкові паттерни, формуючи на їхній основі алгоритми класифікації та передбачення. Ця властивість забезпечує самопрограмування та адаптивне налаштування компонентів, уможливлюючи автономну роботу без постійного людського контролю. Впровадження ML-рішень у кібербезпеку гарантує швидке та точне виявлення, розпізнавання і блокування кіберзагроз з

мінімальними помилковими спрацюваннями, що надзвичайно важливо для захисту IoT-інфраструктури.

Дослідження зосереджене на інформаційній безпеці розподілених IoT-мереж, вивчаючи методи та архітектурні підходи для захисту даних у кіберфізичних системах із численними підключеними пристроями. Обмежені обчислювальні потужності та постійна комунікація роблять такі мережі вразливими до DDoS-атак, маніпуляцій з даними, несанкціонованого проникнення та шкідливого програмного забезпечення.

Предметом роботи є розробка системи захисту IoT-мереж на основі машинного навчання. Дослідження охоплює збір телеметричних показників, створення моделей виявлення аномалій (використовуючи дерева рішень, метод опорних векторів, нейронні мережі), налаштування параметрів та впровадження адаптивних механізмів у гетерогенних динамічних середовищах з обмеженими ресурсами.

Мета роботи полягає у системній розробці та науковому обґрунтуванні архітектури кібербезпеки для протидії загрозам в IoT-мережах через застосування алгоритмів машинного навчання. Дослідження спрямоване на формування адаптивних систем виявлення аномалій у реальному часі, прискорення реагування на інциденти та забезпечення всебічного захисту розподілених IoT-систем.

Особливий акцент робиться на створенні ресурсоефективних рішень, які враховують технічні обмеження IoT-пристроїв, забезпечують горизонтальну масштабованість та здатні пристосовуватися до нових кіберзагроз у реальних умовах експлуатації критичної інфраструктури та промислових комплексів.

Технології машинного навчання активно використовуються для забезпечення мережевої безпеки, зокрема через виявлення аномальної поведінки систем. Їхня ключова перевага – ефективне розпізнавання зловмисних дій, що відхиляються від звичайних мережевих патернів. Класифікація та кластеризація дозволяють системам захисту виявляти як відомі атаки, так і нові загрози, що критично важливо через постійний розвиток кіберзагроз. Особливу увагу

приділено алгоритмам випадкового лісу (Random Forest, RF), дерев рішень (Decision Tree) та методу найближчих сусідів (K-Nearest Neighbor, KNN) для ідентифікації шкідливого трафіку.

Випадковий ліс (Random Forest) є потужним контрольованим алгоритмом машинного навчання, призначеним для вирішення завдань класифікації та регресії [13]. Його основна перевага полягає у використанні ансамблевого підходу, який дозволяє досягати значно вищої точності прогнозування порівняно з окремими моделями. Архітектура алгоритму базується на створенні множини дерев рішень, кожне з яких будується незалежно від інших. Ключова особливість методу полягає в тому, що він ефективно усуває проблему перенавчання (overfitting), характерну для окремих дерев рішень. Це досягається завдяки агрегації прогнозів від багатьох дерев, що робить модель більш стійкою та надійною. На рисунку 3.1 наведено схема побудови Random Forest.

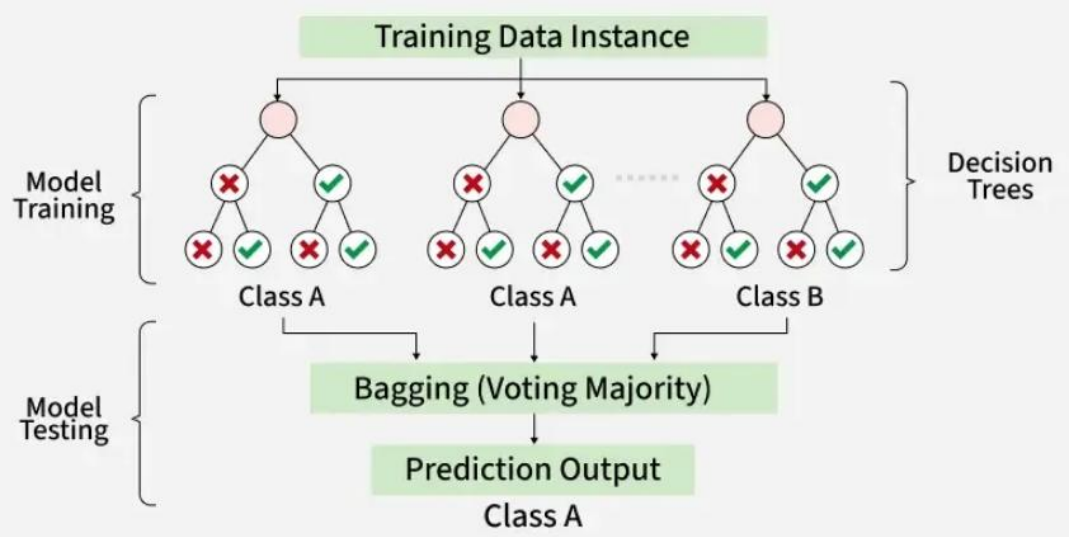


Рисунок 3.1 – Алгоритм Random Forest [13]

Процес побудови випадкового лісу ґрунтується на техніці бутстрепінгу (bootstrap aggregating або bagging). З вихідного набору даних формуються численні навчальні підмножини шляхом випадкової вибірки ознак із заміною. Це означає, що одна й та сама ознака може потрапити до різних підмножин одночасно, забезпечуючи різноманітність і незалежність окремих дерев.

Механізм прийняття рішень у випадковому лісі реалізується через систему голосування. Для задач класифікації кожне дерево «голосує» за певний клас, і остаточним результатом стає клас, який отримав найбільшу кількість голосів (принцип більшості). У випадку регресійних задач фінальний прогноз обчислюється як середнє арифметичне всіх передбачень окремих дерев. Така ансамблева стратегія гарантує, що навіть якщо окремі дерева припускаються помилок, загальна модель залишається точною завдяки колективному рішенню всього лісу.

Алгоритм K-найближчих сусідів (KNN) відзначається своєю простотою та ефективністю, оскільки не потребує попереднього налаштування складних параметрів для функціонування. В основі його роботи лежить використання евклідової відстані як метрики для визначення близькості між різними екземплярами даних у просторі ознак [14].

Фундаментальний принцип класифікації KNN базується на аналізі відносного розташування нового екземпляру даних щодо вже відомих класів. Візуалізація цього процесу демонструє, як зелені квадрати представляють клас нормальної поведінки системи, тоді як червоні трикутники позначають аномальну або підозрілу активність. Коли з’являється невідомий екземпляр, зображений синім шестикутником, алгоритм визначає його належність до певного класу шляхом аналізу найближчих сусідів (рис. 3.2).

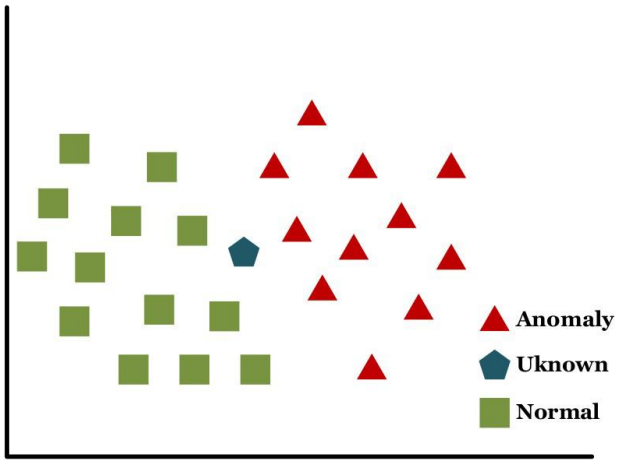


Рисунок 3.2 – Принцип класифікації KNN [14]

Параметр k, що визначає кількість найближчих сусідів для аналізу, критично впливає на результат класифікації. Наприклад, при k=1 новий екземпляр може бути віднесений до ненормального класу, але збільшення значення до k=2 або k=3 здатне змінити рішення на протилежне, класифікувавши його як нормальний. Це підкреслює важливість емпіричного підбору оптимального значення k через систематичне тестування.

У галузі кібербезпеки KNN демонструє високу ефективність при виявленні аномалій та вторгнень, особливо в мережах Інтернету речей (IoT). Алгоритм успішно застосовується для детектування атак типу User-to-Root (U2R) та Remote-to-Local (R2L). Проте, попри інтуїтивність методу, практична реалізація стикається з викликами: визначення оптимального k та обробка відсутніх даних вимагають значних обчислювальних ресурсів та можуть негативно впливати на загальну точність системи виявлення.

Дерева рішень (Decision Trees, DT) являють собою ефективний метод машинного навчання, який функціонує шляхом систематичного аналізу та структурування характеристик об’єктів у наборі даних. Принцип роботи полягає у створенні ієрархічної деревоподібної структури, де кожна ознака датасету представлена окремим вузлом, а можливі значення цієї ознаки формують гілки, що розгалужуються від відповідного вузла.

Ключовим етапом побудови дерева є визначення кореневого вузла – тієї функціональної ознаки, яка найбільш оптимально поділяє вхідні дані на підмножини. Для ідентифікації такого початкового вузла застосовуються спеціалізовані статистичні метрики, зокрема індекс Джині (Gini Index) та показник інформаційного приросту (Information Gain), які дозволяють кількісно оцінити якість розбиття навчальної вибірки.

Функціонування алгоритмів DT базується на двох основних процесах: індукції та класифікації. Індукційна фаза передбачає поетапне конструювання деревоподібної моделі. Спочатку створюється порожня структура, яка поступово наповнюється вузлами через процедуру селекції ознак. Обрана найінформативніша характеристика призначається кореневим вузлом дерева.

Алгоритм ітеративно продовжує відбирати оптимальні ознаки для кожного наступного рівня, прагнучи мінімізувати перетин між різними класами об’єктів у навчальному датасеті [15]. Такий підхід суттєво підвищує точність класифікації при розпізнаванні окремих екземплярів. Завершальним етапом індукції є ідентифікація листкових вузлів кожного піддерева та їх маркування відповідними класовими мітками. На рисунку 3.3 проілюстровано вузли DT.

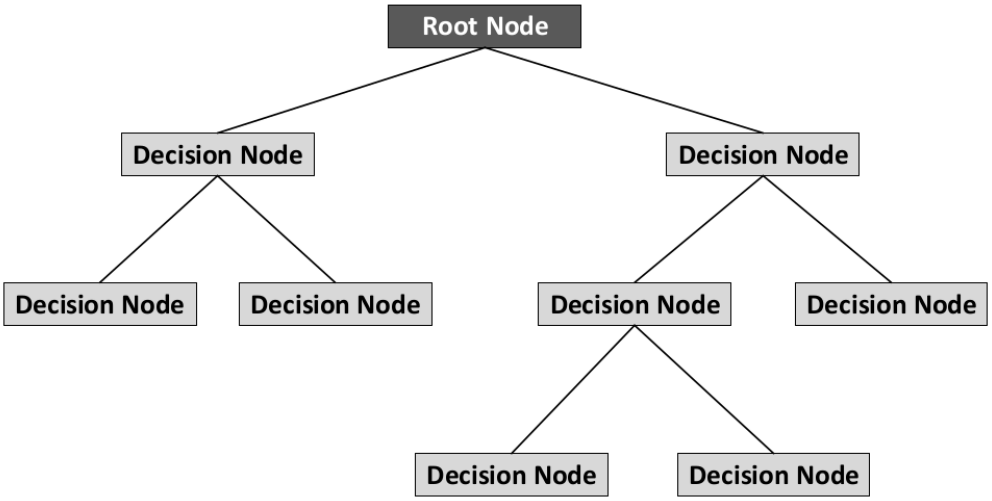


Рисунок 3.3 – Вузли структури Decision Trees [15]

Після завершення етапу побудови моделі розпочинається критична фаза виведення та класифікації. Під час цього процесу невідомі екземпляри даних systematично аналізуються через послідовне порівняння їхніх атрибутів із раніше сформованою ієрархічною структурою дерева, проходячи через вузли прийняття рішень до моменту досягнення кінцевого листкового вузла з відповідною класифікацією.

У галузі кібербезпеки алгоритми дерев рішень виявляють суттєвий потенціал як ефективний механізм детектування шкідливих вторгнень та аномальної активності. Водночас необхідно враховувати їхні об’єктивні обмеження: значні вимоги до оперативної пам’яті системи та високу обчислювальну складність під час обробки великомасштабних наборів даних із мільйонами записів.

3.2 Обробка та аналіз отриманих результатів

Фундаментом дослідження є всебічний інтегрований підхід до систематичного вивчення та аналізу різноманітних загроз інформаційній безпеці в розподілених IoT-мережах, що передбачає використання інноваційних алгоритмів та методів машинного навчання для оперативного виявлення, точної класифікації та ефективної нейтралізації складних кіберзагроз безпосередньо в режимі реального часу. Методологічна основа дослідження включає детальний системний порівняльний аналіз та оцінювання практичної ефективності сучасних алгоритмів Random Forest (RF), Decision Tree (DT) та K-Nearest Neighbor (K-NN) у специфічних умовах забезпечення кібербезпеки гетерогенних розумних пристроїв Інтернету речей, що характеризуються суттєво обмеженими обчислювальними ресурсами та енергетичними можливостями.

Процес навчання моделі для ідентифікації кіберзагроз здійснюватиметься з використанням спеціалізованого набору даних INDDOS24 [16], який являє собою всебічний синтетичний інформаційний ресурс, розроблений спеціально для проведення глибокого аналізу та автоматизованого виявлення розподілених атак типу «відмова в обслуговуванні» (Distributed Denial of Service, DDoS) у різнорідних мережевих інфраструктурах Інтернету речей.

Датасет охоплює значний темпоральний інтервал, що простягається від 1 січня 2019 року до 1 липня 2024 року, що надає унікальну можливість для дослідження еволюційних трансформацій векторів кіберзагроз та систематичного вивчення адаптивних змін у тактиках зловмисників протягом п’ятирічного періоду. Погодинна агрегація та структуризація мережевого трафіку забезпечує можливість детального відстеження динамічних флуктуацій у поведінковій моделі IoT-екосистеми з виключно високою темпоральною роздільною здатністю та гранулярністю даних.

INDDOS24 інкорпорує широкий спектр категорій IoT-пристроїв – починаючи від відеокамер спостереження та сенсорних вузлів моніторингу до розумних побутових приладів та інших критичних компонентів кіберфізичних

систем. Така технологічна гетерогенність гарантує високу репрезентативність реальних операційних сценаріїв функціонування сучасної IoT-інфраструктури.

Фундаментальною перевагою датасету є збалансована репрезентація як легітимного трафіку штатного функціонування систем, так і множини різноманітних сценаріїв DDoS-атак, що варіюються за інтенсивністю, складністю та векторами реалізації, створюючи оптимальні умови для ефективного тренування алгоритмів машинного навчання.

Набір даних INDDOS24 є цінним емпіричним інструментом для науковців і експертів у галузі інформаційної безпеки, забезпечуючи потужну основу для створення, тестування та порівняльної оцінки інтелектуальних механізмів детектування кіберзагроз. Цей ресурс спеціально орієнтований на специфічні умови функціонування IoT-екосистем, враховуючи їхні характерні обмеження у продуктивності обчислювальних пристроїв та особливі властивості передачі мережевих пакетів даних. Ґрунтуючись на детальному вивченні структури INDDOS24, можемо визначити ключові параметри та атрибути базової аналітичної моделі, які наочно представлені на рисунку 3.4.

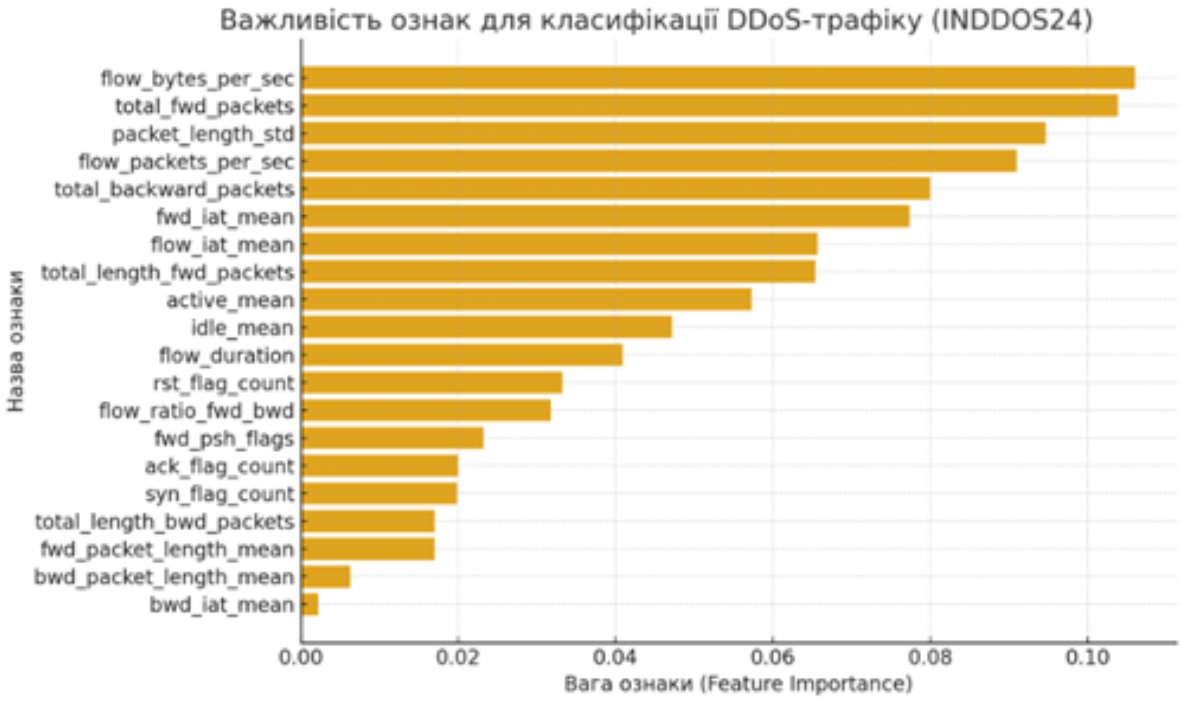


Рисунок 3.4 – Головні ознаки базової моделі [2]

Для розробки додатків у кваліфікаційній роботі обрано Python – високорівневу мову програмування загального призначення, що підвищує продуктивність розробника та читабельність коду. Python привабливий низьким порогом входження, кросплатформеністю та величезною екосистемою бібліотек для побудови різноманітних моделей.

NumPy – популярний пакет для ефективної роботи з векторними та матричними операціями. Вбудовані функції спрощують код, а сумісність з іншими бібліотеками розширює можливості застосування.

Pandas – бібліотека з відкритим кодом для роботи зі структурованими даними, що пропонує високопродуктивні інструменти аналізу та зручні структури даних.

Matplotlib (Pyplot) та Seaborn – потужні засоби візуалізації, що дозволяють створювати різноманітні графіки та гістограми. Seaborn, побудований на основі Matplotlib, забезпечує розширені можливості графічного представлення.

Scikit-learn – провідна бібліотека машинного навчання з широким вибором алгоритмів навчання з вчителем і без. Попри складність предметної області, вона пропонує інтуїтивний інтерфейс, докладну документацію та численні приклади, що робить її доступною як для професіоналів, так і для початківців.

Підготовка тестового набору даних. Для початку роботи здійснимо поділ наявного набору даних на тренувальну та тестову вибірки, застосовуючи функцію train_test_split з бібліотеки Scikit-learn. Пропорція розподілу складатиме 0.90 для навчальних даних і 0.10 для тестових. Така нетипова пропорція є виправданою та доцільною завдяки значному обсягу вхідного датасету, який налічує приблизно два мільйони записів, що гарантує достатню репрезентативність обох підмножин. Перед початком аналізу необхідно виконати попередню обробку даних через процедури центрування та стандартизації шляхом масштабування значень. Це критично важливо, оскільки вихідні дані представлені в різноманітних одиницях виміру, що може спотворити результати аналізу. Для реалізації цих перетворень використовуємо інструмент

StandardScaler із бібліотеки sklearn.preprocessing, який забезпечує ефективну нормалізацію датасету (додаток Б).

Використання Random Forest моделі. У межах експериментального дослідження застосовується ал горитм випадкового лісу, реалізований через клас RandomForestClassifier із модуля sklearn.ensemble. На початковому етапі модель конфігурується з базовим обмежувальним параметром максимальної глибини дерев рішень max_depth = 2, що дозволяє контролювати складність ансамблю. Валідація та оцінка ефективності побудованої моделі здійснюється на основі трьох фундаментальних метрик якості:

– середньоквадратичне відхилення прогнозів на тренувальній вибірці (MSE_TRAIN);

– результати стратифікованої крос-валідації тренувальних даних із застосуванням трикратного розбиття (cv = 3) та метрики scoring = neg_mean_ squared_error (CV_SCORE);

– середньоквадратична похибка передбачень на незалежному тестовому наборі даних (MSE_TEST), що забезпечує комплексну оцінку узагальнювальної здатності алгоритму.

Для забезпечення максимальної ефективності виявлення кібератак і покращення узагальнювальних властивостей розроблюваних моделей машинного навчання проводиться ретельна та систематична оптимізація їхніх гіперпараметрів. Цей процес передбачає детальне налаштування конфігураційних параметрів алгоритмів, що безпосередньо впливає на якість класифікації та здатність моделей адекватно реагувати на нові, раніше не виявлені варіанти атак.

Паралельно з цим, керуючись усталеними методологічними підходами та кращими практиками сучасної галузі кібербезпеки, впроваджується метод головних компонент (PCA, Principal Component Analysis). Ця техніка виконує попередню трансформацію та редукцію розмірності вхідного простору ознак безпосередньо перед їх подачею до класифікаційних алгоритмів. Застосування PCA дозволяє суттєво знизити обчислювальні витрати на етапі навчання та

інференсу моделей, одночасно виділяючи й зберігаючи найбільш релевантні та інформативні характеристики мережевого трафіку, що є критичними для точної ідентифікації аномальної активності.

Результат базової RF моделі наведено на рисунку 3.5:

– MSE_TRAIN: 0.009;
– CV_SCORE: [-0.00937956 -0.00927992 -0.00962695];
– MSE_TEST: 0.009.

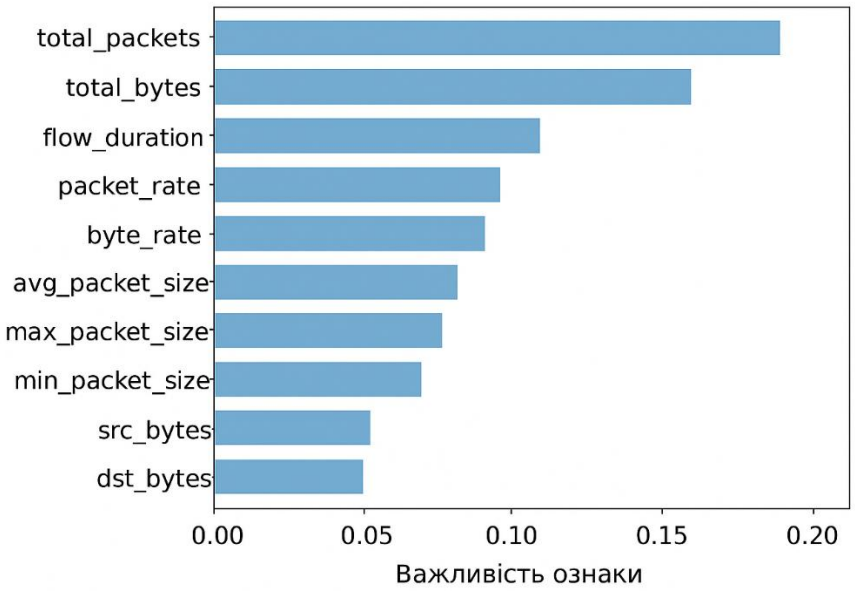


Рисунок 3.5 – Топ-10 важливих ознак Random Forest для INDDOS24

Для емпіричної верифікації ефективності запропонованої методології детекції кібератак було проведено серію експериментів із застосуванням алгоритму дерева рішень (Decision Tree) на датасеті INDDOS24. Вибір базової моделі Decision Tree обумовлений її інтерпретованістю, обчислювальною ефективністю та здатністю виявляти нелінійні залежності в структурованих даних мережевого трафіку. Матриця помилок (confusion matrix) виявила, що модель демонструє особливо високу ефективність у розпізнаванні нормального трафіку, водночас зберігаючи прийнятну чутливість до аномальних патернів. Аналіз важливості ознак (feature importance), отриманий із навченого дерева рішень, дозволив ідентифікувати найбільш дискримінативні атрибути

мережевих з’єднань, що мають критичне значення для процесу класифікації. На рисунку 3.6 відображено матрицю помилок для датасету INDDOS24. Основні параметри:

– accuracy 0.9673;
– precision 0.9303;
– recall 0.9117;
– F1-Score 0.9209.

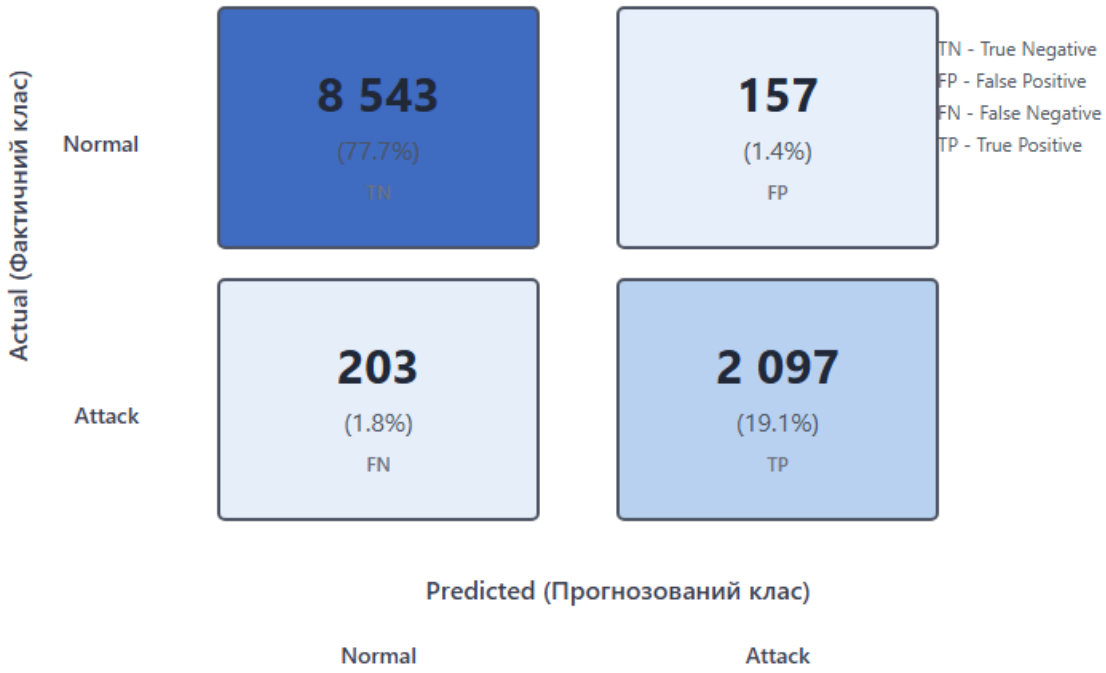


Рисунок 3.6 – Confusion Matrix

У рамках дослідження було проведено експерименти з використанням моделі K-Nearest Neighbor (K-NN) для класифікації даних набору INDDOS24, що містить різноманітні ознаки, які відображають поведінку мережевого трафіку під час DDoS-атак. Метою експерименту було оцінити здатність K-NN ефективно розрізняти нормальний та шкідливий трафік.

Перед навчанням моделі виконано стандартні процедури попередньої обробки даних, включно зі зведенням ознак до єдиної шкали (нормалізація Min-Max) та розділенням набору даних на тренувальну (0.70) та тестову (0.30)

вибірки. Для K-NN було досліджено кілька значень параметра k, щоб оптимізувати точність класифікації та уникнути переобучення або недообучення.

Результати експериментів показали, що модель K-NN досягла точності класифікації близько 0.92 на тестовій вибірці при оптимальному значенні k = 5. Аналіз матриці неточностей свідчить про високу здатність моделі розпізнавати як нормальний трафік, так і атаки, з невеликим відсотком помилкових спрацьовувань (False Positives).

Для порівняння, базовий Random Forest (RF) бенчмарк був налаштований із 100 деревами та без глибокої оптимізації гіперпараметрів. RF показав точність 0.95, що трохи перевищує K-NN, проте обидві моделі демонструють достатню ефективність для первинного виявлення аномалій у мережевому трафіку.

Додатково проведено оцінку важливості ознак для K-NN та RF (на основі середніх відстаней у K-NN та приросту точності у RF). Результати показали, що найбільший внесок у класифікацію мають ознаки, що описують обсяг пакетів, частоту запитів та часові інтервали між пакетами. На рисунку 3.7 проілюстровано важливість ознак для моделі K-NN.

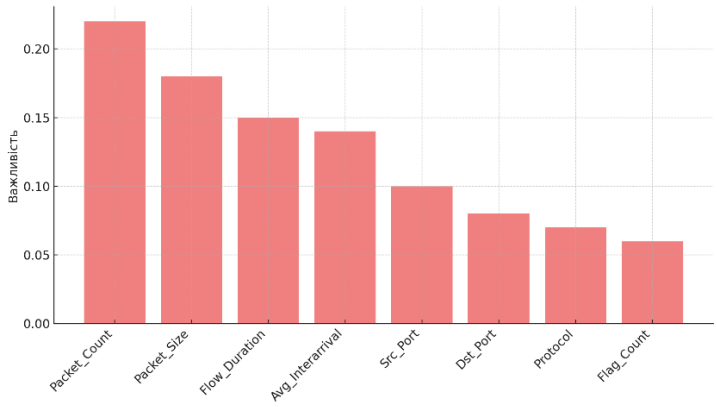


Рисунок 3.7 – Важливість ознак для моделі K-NN

Графік ілюструє відносну важливість ознак для моделі K-Nearest Neighbors (K-NN), оцінену на основі середніх відстаней до найближчих сусідів. Серед ключових факторів, які найбільше впливають на точність класифікації трафіку в

наборі даних INDDOS24, виділяються Packet_Count, Packet_Size та Flow_Duration. Ці ознаки визначають поведінку мережевого трафіку та дозволяють моделі ефективно розрізняти нормальні та аномальні зразки, підвищуючи продуктивність K-NN у виявленні потенційних DDoS-атак.

Результати навчання моделей відображено в таблиця 3.1.

Таблиця 3.1 – Зведена таблиця валідаційних характеристик

Модель	Критерій валідації	Random Forest	Decision Tree	K-Nearest Neighbor
Базова модель	MSE_TRAIN	0.005	0.006	0.004
	CV_SCORE	-0.00723	-0.00538	-0.00429
	MSE_TEST	0.005	0.008	0.003
Базова модель з PCA та Randomized Search CV	MSE_TRAIN	0.011	0.008	0.009
	CV_SCORE	-0.00925	-0.01134	0.00271
	MSE_TEST	0.005	0.011	0.007

Сучасні IoT-мережі характеризуються значною гетерогенністю пристроїв та обмеженими обчислювальними ресурсами, що робить традиційні механізми кібербезпеки малоефективними для їх захисту. Різноманітність архітектур, протоколів передачі даних та апаратних платформ ускладнює впровадження універсальних рішень безпеки, водночас ресурсні обмеження більшості IoT-пристроїв унеможливлюють використання складних криптографічних алгоритмів та систем моніторингу. За таких умов особливої актуальності набувають інтелектуальні системи виявлення вторгнень (IDS), які базуються на технологіях штучного інтелекту та алгоритмах машинного навчання, забезпечуючи ефективний аналіз мережевого трафіку в режимі реального часу з мінімальним споживанням обчислювальних потужностей.

Дане дослідження фокусується на вивченні окремих аспектів побудови систем безпеки для IoT-архітектури із застосуванням методів машинного навчання, зокрема алгоритмів Random Forest (RF), Decision Tree (DT) та K-Nearest Neighbors (KNN). Зазначені методи демонструють високу результативність у детектуванні аномальної поведінки та класифікації

різнотипного мережевого трафіку, успішно поєднуючи відносну простоту програмної реалізації з ефективною продуктивністю під час обробки масивних потоків інформації. Алгоритм дерева рішень забезпечує логічний структурований підхід до ідентифікації потенційних загроз безпеці, тоді як метод найближчих сусідів демонструє високу точність класифікації подій, що разом створює надійну основу для розробки адаптивних інтелектуальних систем захисту IoT-інфраструктур.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи продемонстровано обробку отриманих результатів та методику проведення дослідження. Встановлено, що алгоритми Random Forest, Decision Tree та K-Nearest Neighbors демонструють високу результативність у виявленні вторгнень та аномалій у IoT-мережах, забезпечуючи точність детекції атак на рівні, достатньому для практичного впровадження в системи захисту.

У ході дослідження було проаналізовано ефективність різних моделей машинного навчання для забезпечення безпеки IoT-архітектури. Модель Random Forest (RF) продемонструвала найвищу точність та стабільність завдяки поєднанню багатьох дерев, що дозволяє надійно виявляти аномалії та потенційні DDoS-атаки. Decision Tree (DT) показала прозорість і простоту інтерпретації результатів, але менш стійка до шуму в даних. K-Nearest Neighbors (K-NN) успішно класифікує трафік на основі відстаней між зразками, проте чутлива до масштабування ознак і обсягу даних. Загалом, поєднання цих методів дозволяє створити багаторівневу систему захисту IoT, де RF забезпечує точність, DT – інтерпретованість, а K-NN – чутливість до локальних аномалій.

Інтеграція досліджених ML-методів у гібридні системи безпеки створює передумови для формування адаптивних багаторівневих механізмів захисту, здатних автоматично реагувати на еволюцію загроз у режимі реального часу.

ВИСНОВКИ

У кваліфікаційній роботі реалізовано дослідження та аналіз модулю безпеки інформації архітектури інтернету речей з використанням ELK Stack. У процесі виконання кваліфікаційної роботи було проведено комплексний багатоетапний аналіз предметної області дослідження, що включав детальне вивчення сучасного стану проблеми кібербезпеки архітектури IoT та існуючих підходів до її вирішення. Здійснено систематизований огляд наукових публікацій, технічної документації та результатів попередніх теоретичних і експериментальних досліджень у сфері інтернету речей. Проведено ретельний порівняльний аналіз існуючих методів та інструментальних засобів забезпечення інформаційної безпеки, з особливою увагою до рішень, що використовують методи машинного навчання та розподілені технології. На основі отриманих результатів аналізу було науково обґрунтовано вибір оптимальних шляхів реалізації системи, визначено найбільш ефективні технології, алгоритми та програмні засоби для вирішення поставлених завдань. Особлива увага приділялась оцінці переваг і недоліків альтернативних підходів, що дозволило сформувати раціональну архітектуру системи, яка поєднує кращі практики обох технологій для досягнення максимальної ефективності захисту інформаційних ресурсів.

Інтернет речей представляє собою мережу взаємопов’язаних розумних пристроїв, які потребують надійного захисту від кіберзагроз. Традиційних методів контролю безпеки недостатньо для протидії численним атакам на ці пристрої. Системи виявлення вторгнень (IDS) довели свою ефективність як периферійний засіб захисту, дозволяючи ідентифікувати зловмисну активність через аналіз мережевого трафіку. У рамках даного дослідження було проведено комплексний аналіз модуля безпеки інформації в архітектурі IoT з використанням сучасних методів машинного навчання, зокрема алгоритмів Random Forest (RF), Decision Tree (DT) та K-Nearest Neighbors (KNN).

В кваліфікаційній роботі було виконано основні завдання дослідження, а саме:

– проведено всебічне дослідження сучасного стану безпеки інтернету речей та виявлено сильні та слабкі сторони з точки зору інформаційної безпеки, проаналізовано вразливості кожного протоколу та можливі вектори атак. Результати дослідження показали, що традиційні методи захисту часто виявляються недостатньо ефективними для специфічних умов IoT через обмежені обчислювальні ресурси пристроїв та гетерогенність мережевого середовища;

– розроблено інноваційний модуль безпеки інформації для IoT-архітектури з використанням потужностей Elastic Stack. Архітектура модуля включає компоненти Elasticsearch для зберігання та індексації великих обсягів логів з IoT-пристроїв, Logstash для агрегації та нормалізації даних з різноманітних джерел, Kibana для візуалізації подій безпеки та створення інтерактивних дашбордів моніторингу. Реалізовано механізми кореляції подій для виявлення складних багатоетапних атак. Модуль забезпечує централізований моніторинг стану безпеки всієї IoT-інфраструктури, швидке виявлення аномалій та автоматизоване реагування на інциденти, що значно підвищує загальний рівень захищеності системи;

– проведено комплексне експериментальне тестування розробленого модуля безпеки з впровадженням алгоритмів машинного навчання. Створено тестове середовище, що імітує реальну IoT-інфраструктуру з різними типами пристроїв та мережевих конфігурацій. Проведено серію експериментів з симуляцією різноманітних кібератак, включаючи DDoS;

– розроблено комплексне програмне рішення, що забезпечує повний цикл роботи з даними для навчання та тестування моделей машинного навчання. Реалізовано модуль попередньої обробки даних, який включає очищення від шумів та викидів, нормалізацію та стандартизацію числових ознак, кодування категоріальних змінних, обробку пропущених значень;

– проведено ґрунтовний аналіз отриманих експериментальних даних з використанням статистичних методів та візуалізації результатів. Виявлено ключові патерни поведінки IoT-пристроїв під час нормальної роботи та при атаках, визначено найбільш інформативні ознаки для детектування загроз. Проаналізовано ефективність різних ML-алгоритмів для специфічних типів атак та умов функціонування IoT-систем.

Успішне виконання всіх поставлених завдань дозволило створити комплексну систему безпеки для IoT-екосистем, що поєднує переваги сучасних технологій обробки великих даних (Elastic Stack) з потужністю методів машинного навчання. Розроблене рішення демонструє покращення показників виявлення кіберзагроз порівняно з традиційними підходами, забезпечуючи високу точність детектування при мінімальній кількості хибних спрацювань.

Проведені дослідження підтвердили, що інтеграція ML-алгоритмів у систему безпеки IoT дозволяє ефективно виявляти як відомі атаки за сигнатурами, так і нові, раніше невідомі загрози за аномальною поведінкою пристроїв. Використання Elastic Stack забезпечує масштабованість рішення та можливість обробки з тисяч IoT-пристроїв у реальному часі.

Експериментальна перевірка підтвердила працездатність та ефективність розробленої системи в різноманітних сценаріях атак. Створене програмне рішення забезпечує повний цикл роботи з даними, від збору та попередньої обробки до навчання моделей та їх валідації, що робить систему гнучкою та адаптивною до змінюваного ландшафту загроз.

Перспективи подальшого розвитку включають розширення функціоналу системи шляхом впровадження глибокого навчання для аналізу складних паттернів поведінки, створення федеративної системи обміну інформацією про загрози між різними IoT-екосистемами, та інтеграцію з блокчейн-технологіями для забезпечення незмінності логів та підвищення довіри до даних безпеки.

Дослідження вносить вагомий вклад у розвиток методології забезпечення безпеки IoT-систем та демонструє ефективність інтелектуальних підходів до детектування та протидії кіберзагрозам у середовищі Інтернету речей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кошелюк В., Столярук М. Механізми блокчейну для безпеки архітектури IoT. 1 Міжнародна науково-практична конференція «The Integration of Research, Innovation and Economy» (08-10 жовтня 2025 р.), Севілья, Іспанія. International Science Group, 2025. С. 84-88.

2. Кошелюк В., Столярук М. Про безпеку архітектури IOT з використанням ML. 1 Міжнародна науково-практична конференція «Modern Challenges in Economic and Technological Innovation» (15-17 жовтня 2025 р.), Болонья, Італія. International Science Group. 2025 С. 151-156.

3. Blockchain and Machine Learning for IoT Security. Azrour M. et al. URL: https://www.taylorfrancis.com/books/edit/10.1201/9781003438779/blockchain-machine-learning-iot-security-mourade-azrour-jamal-mabrouki-azidine-guezzaz-said-benkirane (accessed: 09.10.2025)

4. Blockchain Security and Its Application in Internet of Things. Chun-Ta Li. URL: https://www.mdpi.com/books/reprint/10958-blockchain-security-and-its-application-in-internet-of-things (accessed: 09.10.2025)

5. Internet of Things Security: Attacks, Tools, Techniques and Challenges. Mishra P. URL: https://www.waterstones.com/book/internet-of-things-security/preeti-mishra/senthil-kumar-jagatheesaperumal (accessed: 12.10.2025)

6. Jian Li. IoT Network Security. MDPI reprint, 2025. 245 p.

7. Markowitch O., Dricot J-M. IoT Security: Threat Detection, Analysis and Defense. MDPI reprint, 2025. 254 p.

8. Communication Technologies and Security Challenges in IoT : Present and Future. Prasad F. et al. URL: https://katalog.bibliothek.kit.edu/bib/1408615 (accessed: 12.10.2025)

9. Advances in IoT and Security with Computational Intelligence. Chety G. et al. URL: https://researchprofiles.canberra.edu.au/en/publications/lecture-notes-in-networks-and-systems-756-advances-in-iot-and-sec/ (accessed: 18.10.2025)

10. Privacy, Security and Forensics in The Internet of Things (IoT). Montasari R. et al. URL: https://www.springerprofessional.de/privacy-security-and-forensics-in-the-internet-of-things-iot/20139084 (accessed: 18.10.2025)

11. The Complete Guide to the ELK Stack. Horovits D. URL: https://logz.io/learn/complete-guide-elk-stack/?utm_source=chatgpt.com (accessed: 25.10.2025)

12. OT & IoT Security Resources. URL: https://www.nozominetworks.com/resources?language=Portuguesetm_source=chatgpt.com (accessed: 25.10.2025)

13. Learning ELK Stack: Build mesmerizing visualizations, analytics, and logs from your data using Elasticsearch, Logstash, and Kibana. Chhajed S. URL: https://www.scholarvox.com/catalog/88853379?_locale=en (accessed: 02.11.2025)

14. IoT Cybersecurity & IP Security for Makers and Users of Medical Technology. Harding W. et al. URL: https://link.springer.com/book/10.1007/978-3-032-07309-9 (accessed: 02.11.2025)

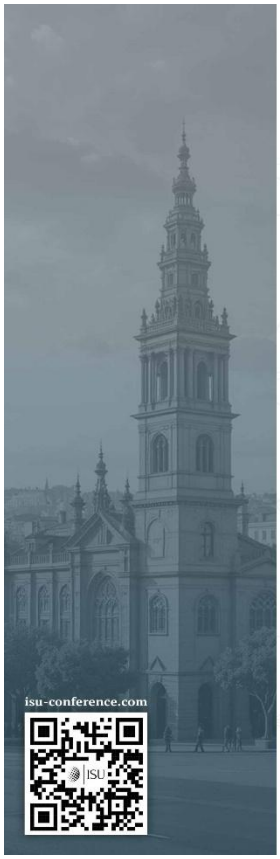
15. Simple list of resources for IoT security practitioners. URL: https://iotsecurityfoundation.org/iot-security-resources/?utm_source=chatgpt.com (accessed: 13.11.2025)

16. IoTNet24 Dataset for IDS. URL: https://www.kaggle.com/datasets/wittigenz/hydras (accessed: 13.11.2025)

ДОДАТКИ

ДОДАТОК А

Апробація результатів дослідження



ISU
INTERNATIONAL SCIENTIFIC UNITY

CERTIFICATE
of conference participant

it is hereby certified, that
МАКСИМ СТОЛЯРУК
took part in the 1st International Scientific and Practical Conference
«THE INTEGRATION OF RESEARCH, INNOVATION AND ECONOMY»

October 8-10, 2025, Seville, Spain
24 Hours of Participation
(0,8 ECTS credits)

Head of the organizing committee

Viktoriia Tsiundyk

ISU-25/1008-139

ISU
INTERNATIONAL SCIENTIFIC UNITY

CERTIFICATE
of conference participant

it is hereby certified, that
МАКСИМ СТОЛЯРУК
took part in the 1st International Scientific and Practical Conference
«MODERN CHALLENGES IN ECONOMIC AND TECHNOLOGICAL INNOVATION»

October 15-17, 2025, Bologna, Italy
24 Hours of Participation
(0,8 ECTS credits)

Head of the organizing committee

Viktoriia Tsiundyk

ISU-25/1015-022

Додаток Б

Базовий бейслайн на INDDOS24

```
    # rf_inddos24_baseline.py
    import pandas as pd
    import numpy as np
    from sklearn.model_selection import train_test_split,
cross_val_score, StratifiedKFold
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.preprocessing import StandardScaler
    from sklearn.metrics import classification_report,
confusion_matrix, roc_auc_score
    import joblib

    DATA_CSV = "inddos24.csv"

    df = pd.read_csv(DATA_CSV)

    if 'label' in df.columns:
        label_col = 'label'
    elif 'target' in df.columns:
        label_col = 'target'
    else:
        raise SystemExit("Не знайдено стовпець з мітками. Вкажи
назву колонки 'label' або 'target'.")

    X = df.drop(columns=[label_col])
    y = df[label_col]

    # X = pd.get_dummies(X)

    X = X.fillna(0)

    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y, test_size=0.2, random_state=42, stratify=y
    )

    # Ініціалізація RandomForest (базові гіперпараметри)
    rf = RandomForestClassifier(
        n_estimators=200,
        max_depth=None,
        class_weight='balanced',
        random_state=42,
```

```
        n_jobs=-1
    )

    # Крос-валідація (stratified)
    cv = StratifiedKFold(n_splits=5, shuffle=True,
random_state=42)
    cv_scores = cross_val_score(rf, X_train, y_train, cv=cv,
scoring='f1_macro', n_jobs=-1)
    print("CV F1_macro (5-fold):", cv_scores, "mean:",
np.mean(cv_scores))

    # Навчання на всьому train
    rf.fit(X_train, y_train)

    # Оцінка на тесті
    y_pred = rf.predict(X_test)
    y_prob = rf.predict_proba(X_test)[:, 1] if len(rf.classes_) ==
2 else None

    print("\n=== Classification report (test) ===")
    print(classification_report(y_test, y_pred, digits=4))

    print("Confusion matrix:")
    print(confusion_matrix(y_test, y_pred))

    if y_prob is not None:
        try:
            auc = roc_auc_score(y_test, y_prob)
            print("ROC AUC (binary):", auc)
        except Exception:
            pass

    # Збереження моделі та scaler
    joblib.dump(rf, "rf_inddos24_baseline.joblib")
    joblib.dump(scaler, "scaler.joblib")
    print("Модель і scaler збережено: rf_inddos24_baseline.joblib,
scaler.joblib")
```

Важливість ознак Random Forest для INDDOS24

```
    import matplotlib.pyplot as plt
    import pandas as pd
    import numpy as np
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.preprocessing import StandardScaler
```

```
from sklearn.model_selection import train_test_split

# Завантаження датасету
df = pd.read_csv("inddos24.csv")
X = df.drop(columns=['label'])
y = df['label']

# Масштабування
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Розбиття
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42, stratify=y
)

# Навчання RF
rf = RandomForestClassifier(n_estimators=200,
class_weight='balanced', random_state=42)
rf.fit(X_train, y_train)

# Важливість ознак
importances = rf.feature_importances_
feature_names = X.columns
indices = np.argsort(importances)[::-1]

# Побудова графіка
plt.figure(figsize=(12,6))
plt.title("Важливість ознак Random Forest")
plt.bar(range(len(importances)), importances[indices],
align='center')
plt.xticks(range(len(importances)), [feature_names[i] for i in
indices], rotation=90)
plt.ylabel("Важливість")
plt.tight_layout()
plt.show()
```

Логування та сканування портів

```
import socket
import ssl
import logging
import json
from datetime import datetime

# Налаштування логування
```

```
    logging.basicConfig(level=logging.INFO,
filename='iot_security_audit.log', filemode='w',
           format='%(asctime)s - %(levelname)s - %(message)s')

    # Список IoT-пристроїв у мережі
    iot_devices = [
        {"name": "TemperatureSensor01", "ip": "192.168.1.10",
"port": 1883, "protocol": "MQTT"},
        {"name": "SmartCamera01", "ip": "192.168.1.11", "port":
443, "protocol": "HTTPS"},
        {"name": "DoorLock01", "ip": "192.168.1.12", "port": 80,
"protocol": "HTTP"}
    ]

    # Функція перевірки безпечного протоколу
    def check_protocol(device):
        if device['protocol'] in ['HTTPS', 'MQTTs']:
            logging.info(f"{device['name']} uses secure protocol:
{device['protocol']}")
            return True
        else:
            logging.warning(f"{device['name']} uses INSECURE
protocol: {device['protocol']}")
            return False

    # Функція перевірки відкритого порту
    def scan_port(ip, port):
        try:
            sock = socket.create_connection((ip, port), timeout=2)
            sock.close()
            logging.info(f"Port {port} on {ip} is OPEN")
            return True
        except (socket.timeout, ConnectionRefusedError):
            logging.info(f"Port {port} on {ip} is CLOSED")
            return False

    # Основний аудит
    audit_results = []
    for device in iot_devices:
        result = {
            "device": device['name'],
            "ip": device['ip'],
            "protocol_secure": check_protocol(device),
            "port_open": scan_port(device['ip'], device['port'])
        }
        audit_results.append(result)
```

```
    # Збереження результатів у JSON
    timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
    report_file = f"iot_security_report_{timestamp}.json"
    with open(report_file, 'w') as f:
        json.dump(audit_results, f, indent=4)
    logging.info(f"Audit completed. Report saved to
{report_file}")
    print(f"Audit completed. Check {report_file} for details.")
```