ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно- інтегрованих технологій, автоматизації, електроінженерії та радіоелектроніки
(повне найменування інституту, назва факультету)

Автоматика та телекомунікації
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____

_____
(підпис) (ініціали, прізвище)

"___"_____20_ р.

Випускна кваліфікаційна робота

бакалавра
(освітньо-кваліфікаційний рівень)

на тему «Розробка стенду з аналізу інформаційної безпеки IoT-мереж»

Виконав: студент 4 курсу, групи ТКР-16
(шифр групи)

спеціальності 172 телекомунікації та радіотехніка
(шифр і назва напряму підготовки, спеціальності)

Удинська Анна Романівна

_____ _____
(прізвище та ініціали) (підпис)

Керівник ст.викл.каф АТ Ступак Г.В. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Покровськ – 2020 р.

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно- інтегрованих технологій, автоматизації, електроінженерії та радіоелектроніки
(повне найменування інституту, назва факультету)

Автоматика та телекомунікації
(повна назва кафедри)

Допустити до захисту:
Декан фКІТАЕР
_____Е.А. Петелін
(підпис та дата)

Захист відбувся_____
(дата)
з оцінкою _____
секретар ДЕК_____
(підпис)

Випускна кваліфікаційна робота

бакалавра
(освітньо-кваліфікаційний рівень)

на тему «Розробка стенду з аналізу інформаційної безпеки IoT-мереж»

Виконав студент групи ТКР-16 Удинська А.Р.
(підпис, дата) (ініціали, прізвище)

Керівник ст.викл.каф АТ Ступак Г.В.
(підпис, дата) (ініціали, прізвище)

Зав. каф. автоматики та телекомунікацій к.т.н., доц. В.В.Поцепаєв
(підпис, дата) (ініціали, прізвище)

Консультанти .
(підпис, дата) (ініціали, прізвище)

(підпис, дата) (ініціали, прізвище)

(підпис, дата) (ініціали, прізвище)

Нормоконтролер ст. викл. Д.О.Жуковська
(підпис, дата) (ініціали, прізвище)

Покровськ – 2020 р.

ДВНЗ «Донецький національний технічний університет»

Факультет комп'ютерно- інтегрованих технологій, автоматизації електроінженерії та радіоелектроніки
Кафедра автоматика та телекомунікації
Освітній ступінь бакалавр
Спеціальність 172 телекомунікації та радіотехніка (шифр і назва)

ЗАТВЕРДЖУЮ:
Завідувач кафедри
_____ /_____/
"____" _____20___року

З А В Д А Н Н Я
НА ВИПУСКНУ КВАЛІФІАЦІЙНУ РОБОТУ СТУДЕНТУ

Удинській Анні Романівні
(прізвище, ім'я, по батькові)Ц

1. Тема роботи «Розробка стенду з аналізу інформаційної безпеки IoT-мереж»
керівник роботи: Ступак Г.В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом від " " року №
2. Строк подання студентом роботи _____
3. Вихідні дані до роботи: результати науково-дослідної роботи, переддипломної практики
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
1) Огляд архітектури Інтернету речей
2) Огляд безпеки та конфіденційності в Інтернеті речей
3) Розгляд типів атак на кожному рівні архітектури IoT
4) Найбільш дієві способи захисту від атак
5) Огляд важливості аутентифікації та контролю доступу в IoT
6) Можливості випробувального стенду для аналізу безпеки в IoT
7) Створення абстрактної моделі архітектури стенду з описом кожного модуля
8) Системна структура та компоненти стенду
9) Опис програмного забезпечення
10) Приклад можливо сценарію тестування
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
1) Абстрактна модель функціональної архітектури стенду
2) Схема компонентів для стенду
3) Приклад звіту одного з можливих сценаріїв тестування

6. Консультанти розділів проекту (роботи)

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

7. Дата видачі завдання_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд архітектури Інтернету речей	25.04.2020	
2	Огляд проблем безпеки та конфіденційності в Інтернеті речей	28.04.2020	
3	Розгляд типів атак на кожному рівні архітектури IoT	30.04.2020	
4	Найбільш дієві способи захисту від атак	04.05.2020	
5	Огляд важливості аутентифікації та контролю доступу в IoT	07.05.2020	
6	Можливості випробувального стенду для аналізу безпеки в IoT	10.05.2020	
7	Створення абстрактної моделі архітектури стенду з описом кожного модуля	13.05.2020	
8	Системна структура та компоненти стенду	18.05.2020	
9	Опис програмного забезпечення	22.05.2020	
10	Приклад можливо сценарію тестування	26.05.2020	
11	Охорона праці	28.05.2020	
12	Оформлення	31.05.2020	

Студент _____ _____
(підпис) (прізвище та ініціали)

Керівник проекту (роботи) _____ _____
(підпис) (прізвище та ініціали)

Лист зауважень

Посада П.І.Б.	Суть зауваження, оцінка та підпис

АНОТАЦІЯ

Удинська А. Р. Розробка стенду з аналізу інформаційної безпеки IoT-мереж / Випускна кваліфікаційна робота на здобуття освітнього ступеня «бакалавр» за спеціальністю 172 телекомунікації та радіотехніка. – ДВНЗ ДонНТУ, Покровськ, 2020.

Робота містить 87 сторінок, складається з вступу, чотирьох розділів, висновків, переліка використаних джерел з 20 найменувань, 27 рисунків, 8 таблиць.

У дипломній роботі розробляється інструментарій тестування безпеки, призначений для IoT-пристроїв. Тестове середовище налаштовано для тестування всіх типів IoT-пристроїв, з різними програмними / апаратними конфігураціями, шляхом проведення тестування безпеки. Для моніторингу роботи тестованого IoT-пристрої в цілому в тестованої платформі використовуються вдосконалені процеси аналізу, засновані на алгоритмах машинного навчання. Розробляється архітектурний проект пропонованого випробувального стенду і опис реалізації.

Ключові слова: ІНТЕРНЕТ РЕЧЕЙ, БЕЗПЕКА, КОНФІНДЕЦІНІСТЬ, МОДЕЛЬ, АНАЛІЗ, АТАКА, ТЕСТ, СТЕНД, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СТРУКТУРА, РИЗИКИ

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

IoT Internet of Things, Інтернет речей – Інтернет речей

USN Ubiquitious Sensor Networks – Всепроникні сенсорні мережі

RFID Radio Frequency Identification - Радіочастотна ідентифікація

IP Internet Protocol –Інтернет-протокол

NFC Near field communication - технологія бездротового високочастотного зв'язку малого радіусу дії

GPS Global Positioning System - Супутникова система навігації

MXO – Мобільні хмарні обчислення

WSN Wireless sensor network - Бездротова сенсорна мережа

PAN Personal Area Network - Персональна мережа

6LoWPAN IPv6 over Low power Wireless Personal Area Networks - Стандарт взаємодії по протоколу IPv6 поверх малопотужних бездротових персональних мереж стандарту IEEE 802.15.4

API Application Programming Interface - Інтерфейс прикладного програмування

TCP Transmission Control Protocol - Протокол управління передачею

UDP User Datagram Protocol - Протокол користувальницьких дейтаграм

SDN Software-defined Networking - Програмно-визначувана мережа

DoS Denial of Service - Відмова в обслуговуванні

DDoS Distributed Denial of Service - Розподілена відмова в обслуговуванні

CSA Cloud Security Alliance - Альянс хмарної безпеки

M2M Machine-to-Machine - Загальна назва технологій, які дозволяють машинам обмінюватися інформацією один з одним, або ж передавати її в односторонньому порядку

IPsec IP Security - Набір протоколів для забезпечення захисту даних, що передаються по міжмережевому протоколу IP

CoAP Constrained Application Protocol - Протокол обмеженого застосування

DTLS Datagram Transport Layer Security - Протокол дейтаграм безпеки транспортного рівня

OS Operating System – Операційна система

XSS Cross-Site Scripting - Міжсайтовий скриптинг

CPU Central Processing Unit - Центральний процесор

CLI Command Line Interface - Інтерфейс командного рядка

SHH Secure Shell - Мережевий протокол прикладного рівня, що дозволяє виробляти віддалене управління операційною системою і тунелювання TCP-з'єднань

SNMD Simple Network Management Protocol - Простий протокол керування мережею

MRM Management and reports module - Модуль управління і звітності

STMM Security testing manager module - Модуль управління тестування безпеки

STM Security testing module - Модуль тестування безпеки

MAM Measurements and analysis module - Модуль вимірювань і аналізу

GUI Graphical User Interface - Графічний інтерфейс користувача

AM Analyzing Mechanism – Аналізуючий механізм

CCM Control And Communication Mechanism – Механізм контролю та зв’язку

CM Coordinating Mechanism – Координаційний механізм

ВСТУП

Через поширення бездротових технологій і міжмашинного обміну, виникнення технології хмарних обчислень і початку переходу на IPv6, отримала розвиток концепція Інтернет речей (Internet of Things, IoT) – концепція, що припускає об'єднання в мережу пристроїв (речей), здатних взаємодіяти один з одним на основі вбудованих технологій, які підтримуються даною мережею. IoT є новим кроком в технологічному прогресі. Інтернет речей дозволяє людям і «речей» з'єднатися в будь-який час і в будь-якому місці, використовуючи різні мережі зв'язку. Основними компонентами IoT є всепроникні сенсорні мережі USN (Ubiquitious Sensor Networks) і радіочастотна ідентифікація RFID (Radio Frequency IDentification) [1]. Інтернет речей ґрунтується на трьох базових принципах:

1. поширена комунікаційна інфраструктура;
2. глобальна ідентифікація об'єктів;
3. можливість кожного об'єкта відправляти і отримувати дані за допомогою персональної мережі або мережі Інтернет, до якої він підключений.

В IoT кожної речі належить свій ідентифікатор, які спільно утворюють систему речей, взаємодіючі одна з одною, створюючи тимчасові або постійні мережі. Так речі можуть брати участь в процесі їх переміщення, ділячись інформацією про поточну геопозицію, що дозволяє повністю автоматизувати процес логістики, а маючи вбудований інтелект, речі можуть змінювати свої властивості і адаптуватися до навколишнього середовища.

Інтернет речей (IoT) складається з комбінації фізичних об'єктів з датчиками, виконавчими механізмами і контролерами з можливістю підключення до цифрового світу через Інтернет. Низька вартість устаткування, поряд з поширеністю мобільних пристроїв і широко поширеним доступом в Інтернет, зробили IoT частиною сучасного повсякденного життя. Інтернет речей є невід'ємною частиною розвитку розумних будинків, розумних міст і розумної системи охорони здоров'я. IoT отримає широке визнання, якщо він

заслужить довіру користувачів, забезпечуючи надійну безпеку і конфіденційність.

Безпека є однією з актуальних тем дослідження сьогодні. Багато дослідників по всьому світу досліджують вирішення різних проблем безпеки в IoT. Однак безпека IoT є великою проблемою через її неоднорідний характер. Інтернет речей є сумішшю дуже багатьох технологій, всі ці технології мають свої власні традиційні недоліки безпеки і конфіденційності, які повинні бути розглянуті в контексті IoT.

На сьогоднішній день 57% IoT-пристроїв схильні до високого ризику кібератак. Кібератаки на ці пристрої можуть привести до серйозних економічних і репутаційних наслідків для будь-якої установи і компанії, що використовує такі пристрої. Кібератака може завдати шкоди бізнесу ще багатьма способами, включаючи витік конфіденційної інформації, крадіжку даних. З огляду на це, важливо захистити свій бізнес, своїх співробітників і своїх клієнтів. Крім того, багато медичних установ використовують IoT, оскільки це полегшує спостереження за пацієнтами. Включено безліч пристроїв, які є частиною IoT, в тому числі кардіостимулятори, дефібрилятори, сканери КТ і т.д. В результаті кібератаки функції цих пристроїв можуть бути ослаблені або повністю відключені. Це може бути дуже небезпечним і призвести до руйнівних наслідків.

Інтернет речей пропонує численні переваги, які можуть полегшити вирішення завдань в нашому повсякденному житті. Мета IoT - забезпечити кращий і простий спосіб роботи і життя. Проте, кожен користувач повинен бути впевнений, що авторизовані люди і машини - єдині, хто має доступ до під'єднаних пристроїв і даним, які вони генерують. Безпека IoT є ключем до завоювання і підтримання довіри клієнтів і до реалізації повного потенціалу пристроїв IoT.

Всі ці факти підтверджують, що існує необхідність серйозного розгляду заходів безпеки для таких пристроїв IoT. Крім того, не існує єдиного стандарту безпеки для всіх пристроїв IoT. Хоча існує потреба у вирішенні проблем

безпеки екосистеми IoT, в даний час не існує гнучкого методу оцінки безпеки пристроїв IoT, і для тестування і аналізу безпеки на пристроях IoT вкрай мало випробувальних заходів.

Розробка випробувального стенду для проведення всебічного тестування і аналізу безпеки пристроїв IoT в реальних умовах може допомогти виправити цю ситуацію. Крім того, через неоднорідність пристроїв IoT (різних типів пристроїв з різними конфігураціями, таких як драйвери пристроїв, апаратні і програмні компоненти і т.д.), потрібний універсальний випробувальний стенд для аналізу безпеки.

В ході виконання дипломної роботи розглянуто детальну архітектуру IoT для кращого розуміння роботи екосистеми IoT, розглянуто основні проблеми безпеки на кожному рівні архітектури, також пропонуються контрзаходи для різних проблем безпеки для захисту систем IoT, обговорюються проблеми, пов'язані із застарілими рішеннями безпеки в IoT.

Також у роботі пропонується розробка випробувального програмного стенду для аналізу безпеки IoT-пристроїв, де представлена абстрактна модель самого стенду у вигляді модульної структури, опис і порівняння пакетів програмного забезпечення, необхідних для проведення аналізу безпеки.

1 ОСНОВНІ ПОНЯТТЯ, АРХІТЕКТУРА ТА СИСТЕМАТИКА INTERNET OF THINGS

Не існує єдиного консенсусу щодо архітектури IoT, який узгоджений універсально. Різні архітектури були запропоновані різними дослідниками [2].

1.1 Трирівнева архітектура та п’ятирівнева архітектури

Основна архітектура [2] - це трирівнева архітектура, як показано на Рисунку 1.1. Вона була введена на ранніх етапах досліджень у цій галузі. Вона має три рівні, а саме: рівень сприйняття, мережевий рівень та прикладний рівень:

1. Рівень сприйняття - це фізичний рівень, який має датчики для обробки та збору інформації про навколишнє середовище. Він чутливий до деяких фізичних параметрів та ідентифікує інші розумні об’єкти в оточенні.

2. Мережевий рівень відповідає за з'єднання з іншими розумними речами, мережевими пристроями та серверами. Його функції також використовуються для передачі та обробки даних датчиків.

3. Прикладний рівень відповідає за доставку користувачу конкретних послуг. Він визначає різні програми, в яких можна використовувати Інтернет речей, наприклад, розумні будинки, розумні міста та розумне здоров'я.

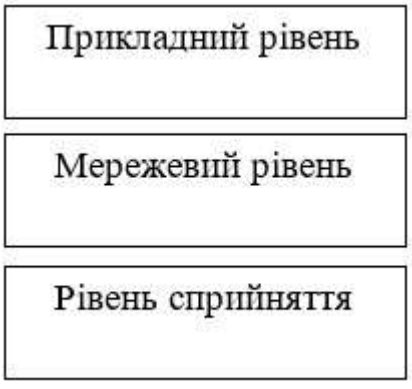


Рисунок 1.1 – Трирівнева архітектура IoT

Трирівнева архітектура визначає основну ідею Інтернету речей, але її недостатньо для досліджень IoT, оскільки дослідження часто зосереджуються на тонших аспектах Інтернету речей. Ось чому існує ще багаторівневих архітектур, запропонованих у різних дослідженнях [3, 4, 5, 6]. Одна з них - це п'ятирівнева архітектура, яка додатково включає рівні обробки та бізнес-рівень. П'ять рівнів - це рівень сприйняття, транспортний рівень, рівень обробки, прикладний рівень та бізнес-рівень (Рисунок 1.2). Роль рівнів сприйняття та прикладного рівня така ж, як і архітектура з трьома рівнями. Функції решти трьох рівнів:

1. Транспортний рівень передає сенсорні дані від рівня сприйняття до рівня обробки і навпаки через бездротову мережу, 3G/4G, локальну мережу, Bluetooth, RFID і NFC.

2. Рівень обробки також відомий як проміжний рівень. Він зберігає, аналізує та обробляє величезну кількість даних, що надходять із транспортного рівня. Він може керувати і надавати різноманітний набір послуг нижчим верствам. Тут використовується багато технологій, таких як бази даних, хмарні обчислення та великі модулі обробки даних.

3. Бізнес-рівень керує всією системою IoT, включаючи додатки, бізнес-моделі та моделі прибутку, а також конфіденційність користувачів.

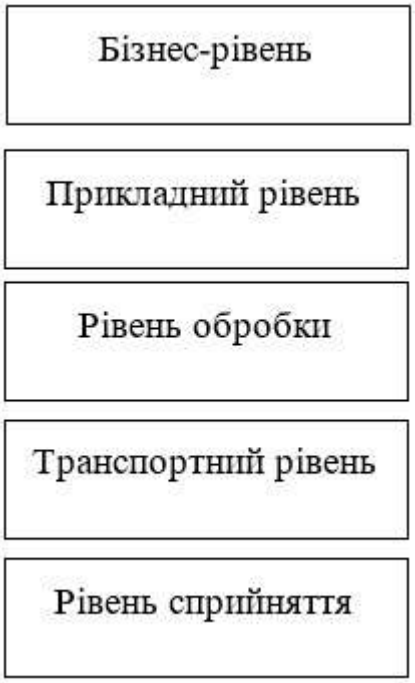


Рисунок 1.2 – П’ятирівнева архітектура IoT

1.2 Систематика IoT

1.2.1 Рівень сприйняття

Рівень сприйняття (Рисунок 1.3) збирає дані за допомогою датчиків, які є найважливішими драйверами Інтернету речей [7]. Існують різні типи датчиків, які використовуються в різних додатках IoT: датчик місцезнаходження (GPS), датчики руху (акселерометр, гіроскоп), камера, датчик світла, мікрофон, датчик близькості та магнітометр. Вони широко використовуються в різних додатках IoT. Починають застосовуватися багато інших типів датчиків, таких як датчики для вимірювання температури, тиску, вологості, медичних параметрів організму, хімічні та біохімічні речовини та нейронні сигнали.

Інший клас датчиків - це інфрачервоні датчики. Зараз вони широко використовуються в багатьох додатках IoT: IR-камерах, детекторах руху, вимірюванні відстані до об'єктів, що знаходяться поблизу, наявності диму та газів, а також як датчики вологи. Усі програми IoT повинні мати один або більше датчиків для збору даних з навколишнього середовища.

Датчики - важливі компоненти розумних об'єктів. Одним з найважливіших аспектів Інтернету речей є усвідомлення контексту, що неможливо без сенсорної технології. Датчики IoT в основному мають невеликі розміри, мають низьку вартість і споживають менше енергії. Їх обмежують такі фактори, як ємність акумулятора та простота розгортання.

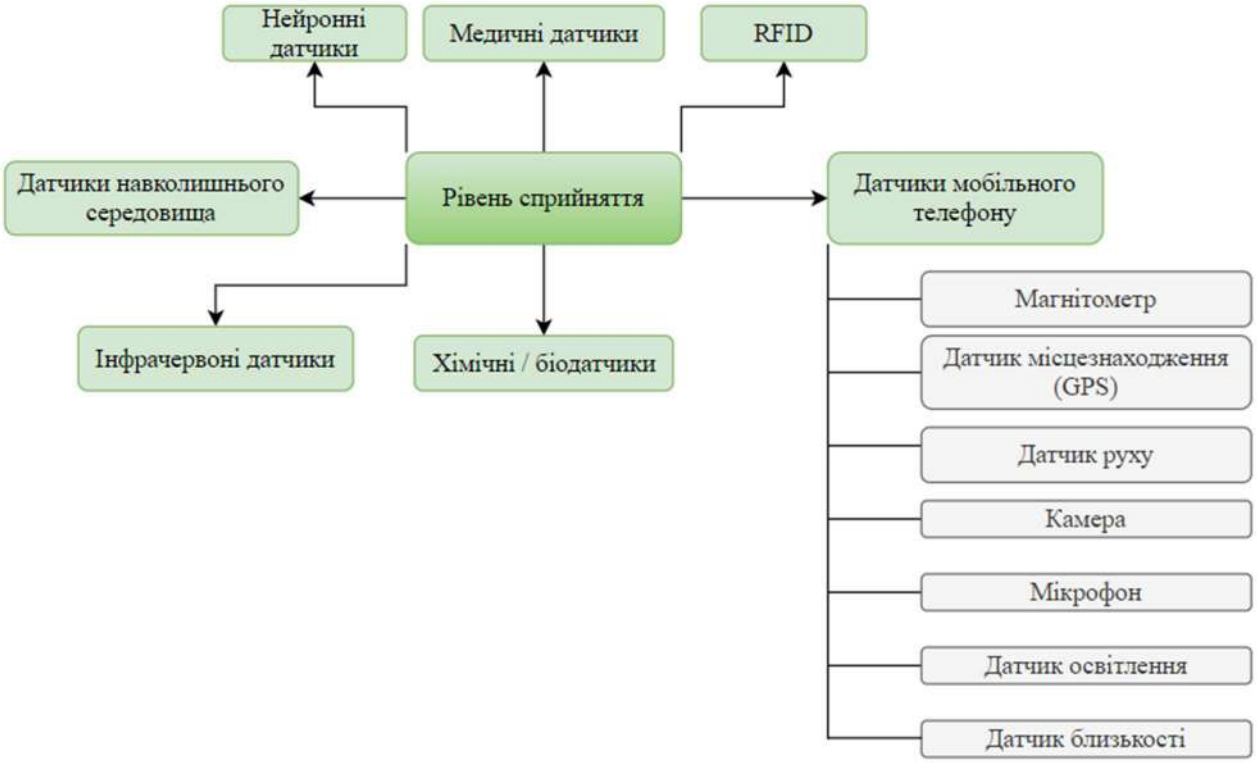


Рисунок 1.3 – Рівень сприйняття в систематиці IoT

1.2.2 Рівень обробки

Оскільки розумні речі збирають величезну кількість сенсорних даних, для аналізу, зберігання та обробки цих даних необхідні обчислення та зберігання ресурсів. Найпоширеніші ресурси для обчислення та зберігання - це хмарні, оскільки вони пропонують велику обробку даних, масштабованість та гнучкість. Але цього буде недостатньо для задоволення вимог багатьох застосувань IoT через наступні причини [8]:

1. Мобільність: більшість розумних пристроїв мобільні. Їхнє мінливе розташування ускладнює спілкування з хмарним центром обробки даних через зміни мережевих умов у різних місцях.

2. Надійне спрацьовування в режимі реального часу: для спілкування з хмарним середовищем та отримання відповідей потрібен час. Програми, залежні від затримок, які потребують реальної відповіді в реальному часі, можуть виявитися недоцільними для цієї моделі. Також зв’язок може бути втратним через бездротові посилання, що може призвести до недостовірних даних.

3. Масштабованість: більше пристроїв означає більше запитів до хмарного середовища, тим самим збільшуючи затримку.

4. Обмеження живлення: комунікація споживає багато енергії, а пристрої IoT працюють від акумулятора. Таким чином, вони не можуть дозволити собі працювати постійно.

Для вирішення проблеми мобільності дослідники запропонували мобільні хмарні обчислення (МХО) [9]. Але все ж є проблеми, пов’язані із затримкою та потужністю. МХО також страждає від проблем мобільності, таких як часто мінливі мережеві умови, через які виникають такі проблеми, як згасання сигналу та деградація послуги.

Для вирішення цих проблем можна привести деякі ресурси для обчислення та зберігання на так званий «край мережі», а не покладатися на хмарні обчислення. Ця концепція відома як туманні обчислення [10]. Туманні обчислення можна розглядати як хмарні: дані можна зберігати, обробляти, фільтрувати та аналізувати на «краю мережі», перш ніж надсилати їх у «хмару» через дорогі засоби зв'язку. Парадигми туманних та хмарних обчислень йдуть разом. Обидва вони необхідні для оптимальної роботи програм IoT. Розумний шлюз [11] може бути використаний між базовими мережами та хмарою, щоб реалізувати туманні обчислення, як показано на Рисунку 1.4.

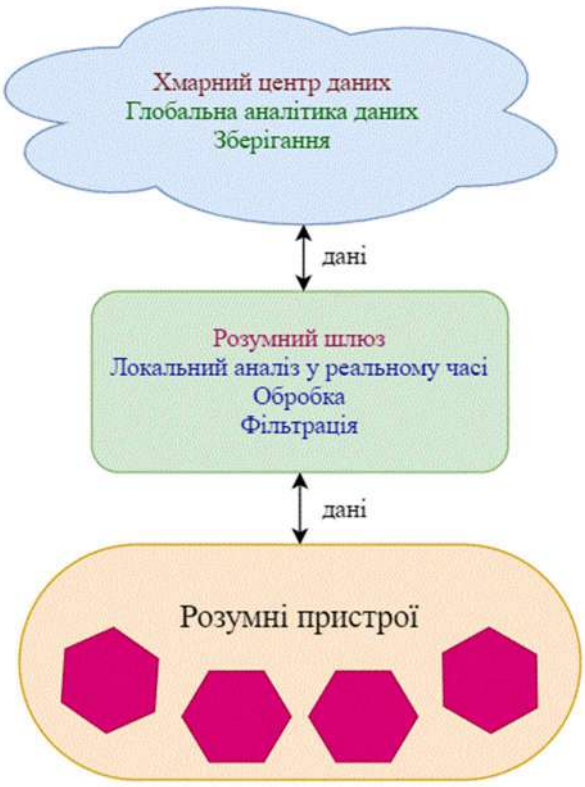


Рисунок 1.4 – Використання розумного шлюзу для туманних обчислень

1.2.3 Мережевий рівень

Різними технологіями на цьому рівні є мобільні мережі, супутникові мережі, бездротова спеціальна мережа та безліч захищених протоколів зв’язку, які використовуються в цих технологіях(Рисунок 1.5).

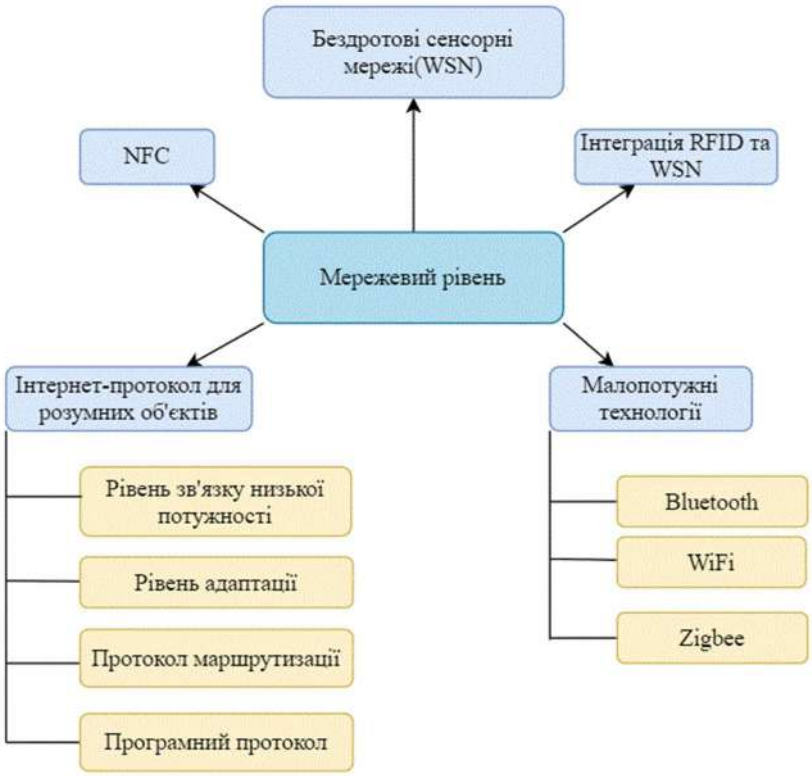


Рисунок 1.5 – Мережевий рівень в систематиці IoT

Оскільки Інтернет речей дуже швидко зростає, існує велика кількість різнорідних розумних пристроїв, що підключаються до Інтернету. Пристрої IoT працюють з мінімальними ресурсами для обчислення та зберігання. Через їх обмежений характер пов'язані різні комунікаційні проблеми, які є наступними [12]:

1. Адресація та ідентифікація: оскільки мільйони розумних речей будуть підключені до Інтернету, їх доведеться ідентифікувати за допомогою унікальної адреси, на основі якої вони спілкуються між собою. Для цього потрібен великий простір адрес та унікальна адреса для кожного розумного об’єкта;

2. Зв'язок з низькою потужністю: передача даних між пристроями - це енергоємне завдання, зокрема, бездротовий зв'язок. Тому потрібне рішення, яке полегшує зв’язок із низьким споживанням енергії;

3. Протоколи маршрутизації з низькою потребою в пам'яті та ефективними схемами зв'язку;

4. Висока швидкість втрат зв'язку;

5. Мобільність розумних речей.

Пристрої IoT зазвичай підключаються до Інтернету через стек TCP/IP (Рисунок 1.6) . Цей стек дуже складний і вимагає великої кількості енергії та пам'яті від підключених пристроїв. Пристрої IoT також можуть локально підключатися через мережі, що не стосуються TCP/IP, які споживають менше енергії, та підключатися до Інтернету за допомогою смарт-шлюзу.

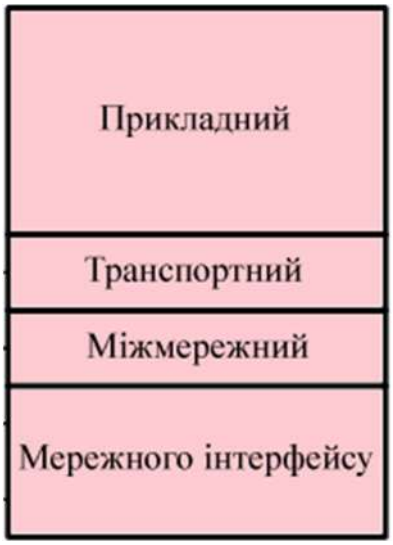


Рисунок 1.6 – Концептуальні рівні стека TCP/IP

Канали зв'язку, які не належать до TCP/IP, такі як Bluetooth, RFID та NFC є досить популярними, але обмежені в своєму діапазоні (до кількох метрів). Тому їх застосування обмежується невеликими персональними мережевими зонами. Приватні мережі (PAN) широко використовуються в додатках IoT, таких як носіння, підключення до смартфонів. Для збільшення діапазону таких локальних мереж виникла потреба змінити стек TCP/IP, щоб полегшити зв'язок з низькою потужністю за допомогою TCP/IP стека. Одне з рішень - 6LoWPAN, який включає IPv6 з персональними мережами малої потужності. Діапазон PAN з 6LoWPAN аналогічний локальним мережам, а споживання електроенергії значно нижче.

1.2.4 Проміжний рівень

Повсюдні обчислення є ядром "Інтернету речей", що означає включення обчислень і підключення до всього, що нас оточує. Взаємодія таких різнорідних пристроїв вимагає чітко визначених стандартів. Але стандартизація ускладнена через різноманітні вимоги, що пред'являються до різних програм і пристроїв. Для таких гетерогенних додатків рішенням є наявність проміжної програмної платформи, яка буде абстрагувати деталі речей для додатків. Вона повинна діяти як програмний міст між речами і додатками. Вона повинна надавати необхідні послуги розробникам додатків [13], щоб вони могли більше зосередитися на вимогах додатків, а не на взаємодії з базовим обладнанням. Підводячи підсумок, проміжне програмне забезпечення абстрагує апаратне забезпечення і надає інтерфейс прикладного програмування (API) для зв'язку, управління даними, обчислень, безпеки і конфіденційності (Рисунок 1.7).

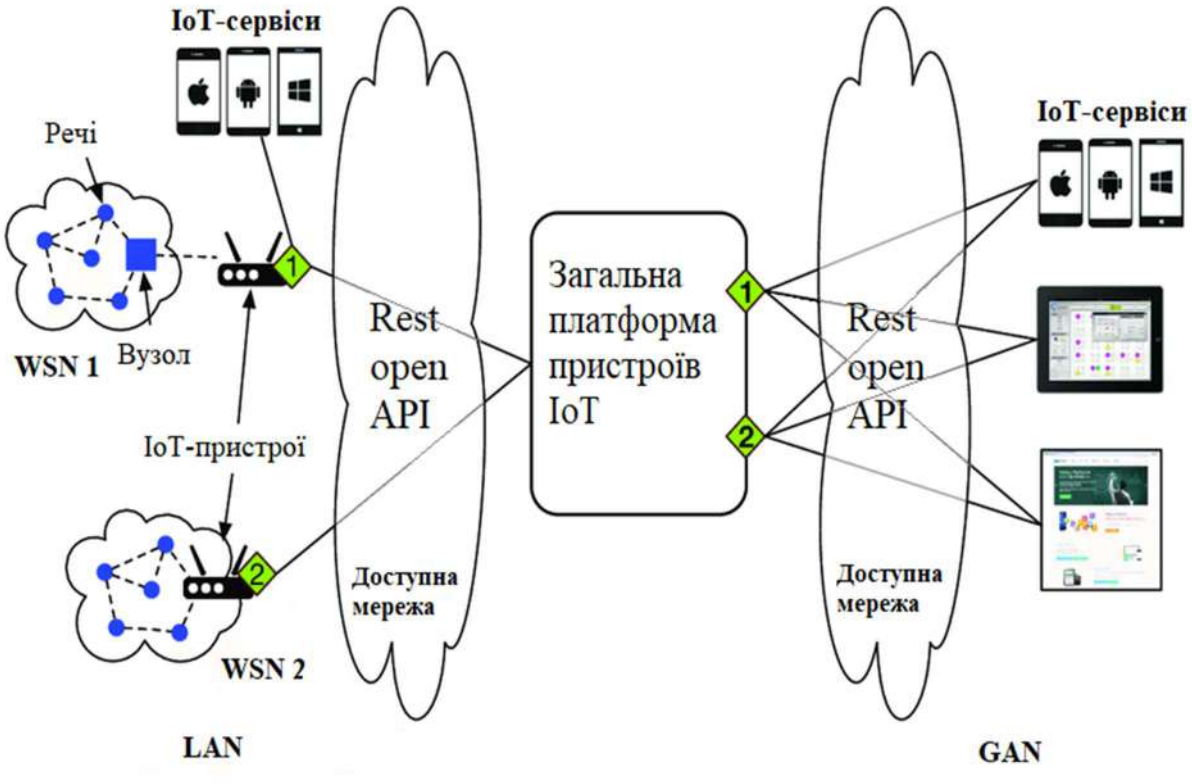


Рисунок 1.7 – Проміжна програмна платформа API

Існує багато проміжних програмних рішень для Інтернету речей, які вирішують такі проблеми як: сумісність та абстрагування програм; відкритість

та управління пристроями; масштабованість; робота з великими даними та аналітика; безпека та приватність; робота з хмарними сервісами. Для Інтернету речей існує багато рішень проміжного програмного забезпечення, які стосуються однієї або декількох згаданих вище проблем. Усі вони підтримують сумісність та абстракцію, що є головною вимогою середнього програмного забезпечення. Програмна продукція може бути класифікована як показано на Рисунку 1.8.

Рисунок 1.8 – Проміжний рівень в систематиці IoT

1.2.5 Прикладний рівень

Існує цілий ряд областей, в яких були розроблені інтелектуальні програми (рис 1.7). Попередні дослідження показують потенціал IoT в поліпшенні якості життя в нашому суспільстві. Деякі застосування IoT додатків використовуються в домашній автоматизації, стеженні за фітнесом,

моніторингу здоров'я, охороні навколишнього середовища, розумних містах і промислових умовах.

Рисунок 1.9 – Прикладний рівень в систематиці IoT

1.3 Особливості проектування у системі IoT

1.3.1 Проектування датчиків

Навіть незважаючи на те, що вибору щодо датчиків може бути не так вже й багато, безумовно є великий вибір щодо можливостей обробки та роботи в мережі, які об'єднані разом з датчиками. Вибір варіюється від невеликих Sub-mW плат, до Arduino або Atom плат, які споживають 300-500 мВт потужності. Цей вибір залежить від ступеня аналітики і попередньої обробки даних, яку ми хочемо виконати на самому датчику (Рисунок 1.10).

Існує також питання логістики. Для створення sub-mW-плати потрібна експертиза проектування плати, а це може бути не завжди доступно. Тому, можливо, було б доцільно встановити датчик в комплекті з комерційно доступними комплектами вбудованих процесорів.

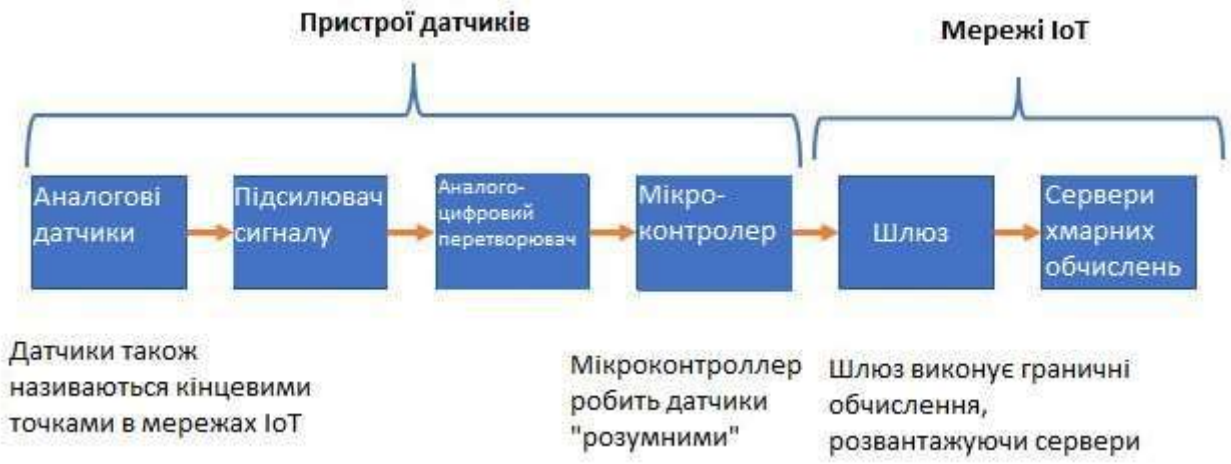


Рисунок 1.10 – Проектування пристроїв системи датчиків в IoT

1.3.2 Комунікація

У вузлах IoT живлення є найбільш домінуючим питанням. Живлення, необхідне для передачі і прийому повідомлень, становить більшу частину від загальної потужності, і в результаті вибір мережевої технології є життєво важливим. Важливими чинниками, які повинні враховуватись, є відстань між відправником і отримувачем, характер перешкод, спотворення сигналу, навколишній шум, а також урядові постанови. На Рисунку 1.11 зображено рекомендовані мережеві технології залежно від діапазону та типу мережі.

Грунтуючись на цих ключових факторах, необхідно вибрати конкретний протокол бездротової мережі. Наприклад, якщо потрібно просто спілкуватися всередині невеликої будівлі, можна використовувати Zigbee, в той час, якщо потрібен зв'язок в "розумному" місті, вибір іде в сторону Sigfox або LoraWAN. До того ж часто існують значні обмеження на частоту і потужність, яка може бути витрачена на передачу. Ці обмеження в основному накладаються державними органами. Необхідно прийняти правильне рішення, беручи до уваги всі ці фактори.

Рисунок 1.11 – Використання рекомендованих мережевих технологій залежно від діапазону та типу мережі

1.3.3 Проміжне програмне забезпечення

Перший вибір, який необхідно зробити, це вибрати між проміжним програмним забезпеченням з відкритим вихідним кодом, таким як FiWare, або пропрієтарним рішенням. І те, і інше має свої плюси і мінуси. В теорії проміжне програмне забезпечення з відкритим вихідним кодом більш гнучке; однак, воно може мати обмежену підтримку IoT-пристроїв. В ідеалі необхідно, щоб проміжне програмне забезпечення взаємодіяло з усіма типами комунікаційних протоколів і пристроїв: однак це може бути не так.

Отже, якщо потрібна сувора сумісність з певними пристроями та протоколами, то пропрієтарне рішення краще. Проте, пропозиції з відкритим вихідним кодом мають цінові переваги і іноді їх простіше впровадити. Деякі приклади поширеного програмного забезпечення наведено у Таблиці 1.1.

Таблиця 1.1 – Приклади деяких поширених проміжних програмних забезпечень для IoT

Назва	Опис

AllSeen Alliance (AllJoyn)	Платформа взаємодії AllJoyn, контрольована AllSeen Alliance (ASA), є найбільш широко використовуваною платформою IoT з відкритим вихідним кодом.
Bug Labs dweet and freeboard	Bug Labs пропонує платформу для обміну повідомленнями та оповіщенням «dweet» і «безкоштовний додаток» для розробки IoT. Dweet допомагає публікувати та описувати дані за допомогою WEB-API HAPI і JSON. Freeboard - це інструмент перетягування для проектування інформаційних панелей і візуалізації IoT.
DSA	Архітектура розподілених послуг полегшує децентралізоване взаємодія пристроїв, логіку і додатки. Проект DSA створює бібліотеку розподілених службових зв'язків (DSLinks), які дозволяють трансляцію протоколу і інтеграцію даних зі сторонніми джерелами. DSA пропонує масштабовану топологію мережі, що складається з декількох DSLink, що працюють на прикордонних пристроях IoT, підключених до багаторівневої ієрархії брокерів.
Kaa	Проект Kaa, підтримуваний CyberVision, пропонує масштабовану наскрізну інфраструктуру IoT, призначену для великих мереж IoT, підключених до хмари. Платформа включає в себе серверну функцію з підтримкою REST для служб, аналітики та управління даними, зазвичай розгортається у вигляді кластера вузлів, що координуються Apache Zookeeper.
Macchina.io	Надає середу виконання з підтримкою веб-інтерфейсу, модульну та розширювану середу JavaScript і C ++ для розробки додатків шлюзу IoT, що працюють на хакерських платах Linux.

Також потрібно вибрати протокол взаємодії і переконатися, що він сумісний з брандмауерами відповідних організацій. Загалом, вибір протоколу, заснованого на HTTP, найкращий з цієї точки зору. Також потрібно вибрати між TCP і UDP. UDP завжди краще з точки зору енергоспоживання. Поряд з цими міркуваннями, також потрібно розглянути варіанти зберігання потоків даних датчиків, мови запитів і підтримку генерування динамічних сповіщень.

1.3.4 Прикладний рівень

Більшість фреймворків IoT забезпечують значну підтримку для створення прикладного рівня. Сюди входять API для інтелектуального аналізу даних, обробки даних і візуалізації. Проте, тут потрібен компроміс між наданими

функціями і необхідними ресурсами. Не потрібен дуже важкий фреймворк, якщо немає необхідності мати багато можливостей. Ці проблеми вирішують розробники програми.

1.4 Висновки за Розділом 1

У Розділі 1 докладно розглянуто архітектуру екосистеми IoT, яка складається в основному з п’яти рівнів: рівень сприйняття, рівень обробки, мережевий рівень, проміжний рівень прикладний рівень. Також розглянуто систематику та опис роботи кожного рівня архітектури IoT: принцип роботи, які технології використовуються, за допомогою яких пристроїв та ресурсів відбувається обмін даними. Також розглянуто складність проектування в IoT на кожному рівні та запропоновано вирішення проблем проектування.

2 ІНФОРМАЦІЙНА БЕЗПЕКА ІНТЕРНЕТУ РЕЧЕЙ, АВТЕНТИФІКАЦІЯ ПРИСТРОЇВ ТА КОНТРОЛЬ ДОСТУПУ

2.1 Безпека Інтернету речей

Як вже було сказано раніше, мета IoT – забезпечити кращий і простий спосіб роботи і життя. Але щоб досягти повністю цієї мети, треба бути впевненими у безпеці користування IoT-пристроями та цілісності даних, обмін якими відбувається у мережі. Адже безпека IoT – ключ до завоювання і підтримання довіри клієнтів і до реалізації повного потенціалу пристроїв IoT. Але поки що на сьогодні дві третини таких пристроїв, які схильні до високого ризику кібератак.

Отже, крім величезного значення і широкого застосування IoT, його нелегко розгорнути в критично важливих прикладних областях, де безпека і конфіденційність є найбільш важливими проблемами. Наприклад, успішний напад на інтелектуальну систему охорони здоров'я може привести до загибелі багатьох людей, а також до фінансових втрат і людських жертв в разі інтелектуальної транспортної системи. Безпека IoT є складною областю і вимагає подальших досліджень для вирішення цих проблем.

Інтернет речей містить у собі суміш різноманітних технологій як на апаратному, так і на прикладному рівні. Тому проблеми безпеки будуть розглянуті на кожному рівні архітектури IoT, яка була приведена в Розділі 1 (Рисунок 2.1).

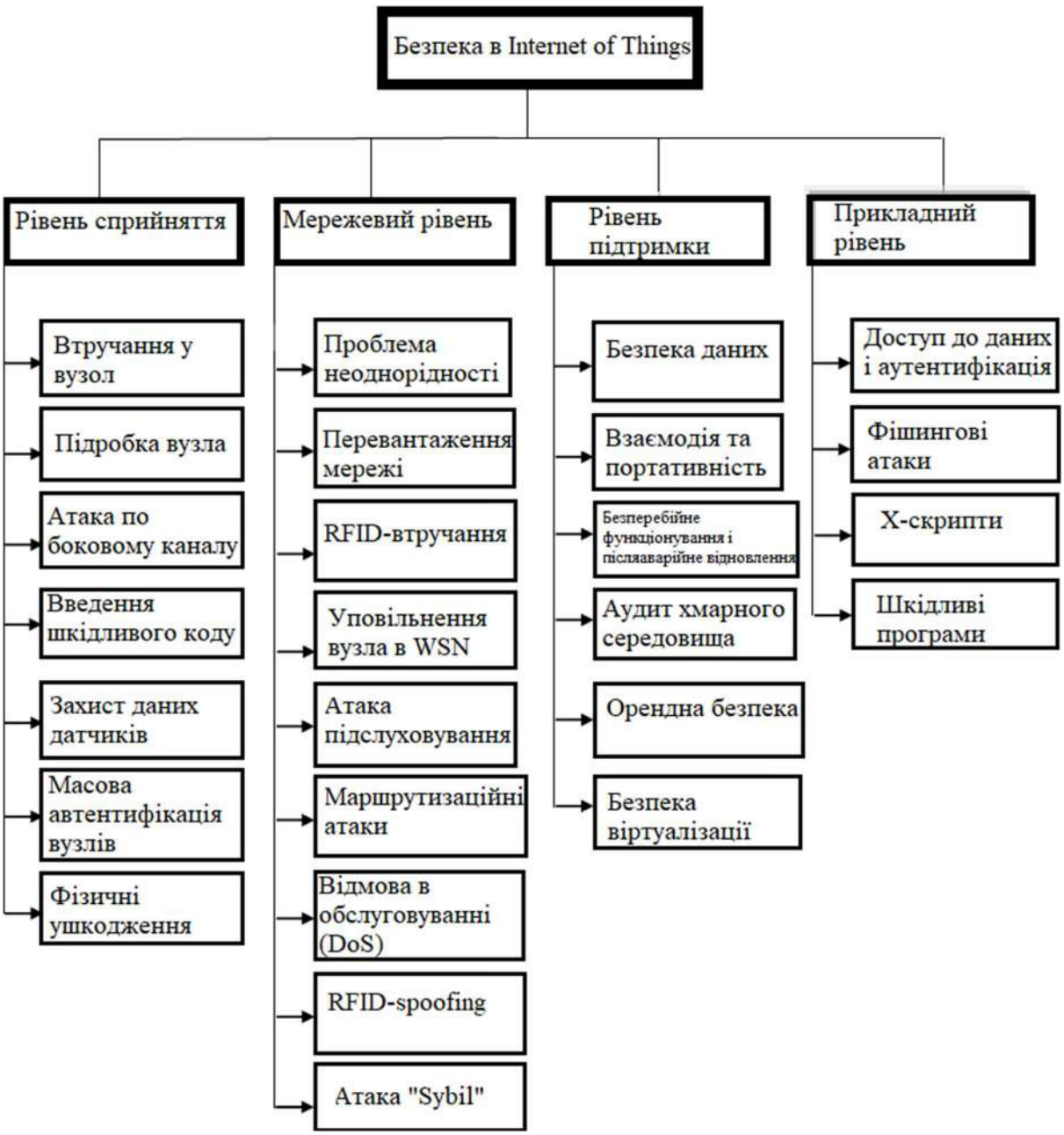


Рисунок 2.1. Проблеми безпеки в архітектурі IoT

2.1.1 Безпека на рівні сприйняття

Рівень сприйняття складається із обмежених ресурсами пристроїв IoT, тобто датчиків, RFID-міток, пристроїв Bluetooth та Zigbee. Ці пристрої більш схильні до кібератак. Оскільки велика кількість пристроїв IoT фізично розгорнута у відкритих областях, вона стикається з багатьма фізичними атаками, якими є:

- Втручання у вузол: якщо атакуючий має фізичний доступ до вузлів датчиків, він може замінити весь вузол або частину його апаратного забезпечення, а також може підключитися безпосередньо до нього, щоб змінити деяку конфіденційну інформацію і отримати доступ до вузла [14]. Чутливою інформацією можуть бути криптографічні ключі або таблиці маршрутизації;

- Підроблення вузла: зловмисник може додати підроблений вузол в систему IoT і впровадити шкідливі дані через цей підроблений вузол в мережу, що зробить пристрої з низьким енергоспоживанням зайнятими і вони будуть споживати більше енергії [15];

- Атака на боковий канал: зловмисники використовують таку інформацію, як споживання енергії, витрата часу та електромагнітне випромінювання від вузлів датчиків для атаки механізмів шифрування [15];

- Фізичні ушкодження: зловмисник може фізично пошкодити пристрій IoT для відмови в обслуговуванні. Пристрої IoT встановлюються як у відкритих, так і в закритих приміщеннях і більш сприйнятливі до фізичного ушкодження з боку зловмисника;

- Введення шкідливого коду: зловмисник фізично компрометує вузол, вставляючи в нього шкідливий код, який дасть йому незаконний доступ до системи [16];

- Захист даних датчиків: вимоги до конфіденційності даних датчика невисокі, тому що противник може помістити датчик поруч з сенсором системи IoT і може відчувати ту ж цінність, проте його цілісність і автентичність більш важливі і повинні бути надійно захищені;

- Масова автентифікація вузлів: велика кількість вузлів в системі IoT стикаються з проблемами аутентифікації [15]. Величезна кількість мережевих з'єднань потрібно тільки для цілей аутентифікації, що впливає на продуктивність.

На сьогоднішній день частіше за все пропонується вирішення таких проблем за допомогою використання програмно-визначеної мережі (Software Defined Networking, SDN). Це забезпечує кращу продуктивність при менших

витратах, а також знижує вартість мережевих ресурсів, що використовуються в мережі. Тому SDN використовується в якості зв'язку з IoT для усунення проблем, пов'язаних з безпекою. Обидві технології об'єднують свої архітектури в одну архітектуру, яка має три пристрої: агент IoT, контролер IoT і контролер SDN (Рисунок 2.2)

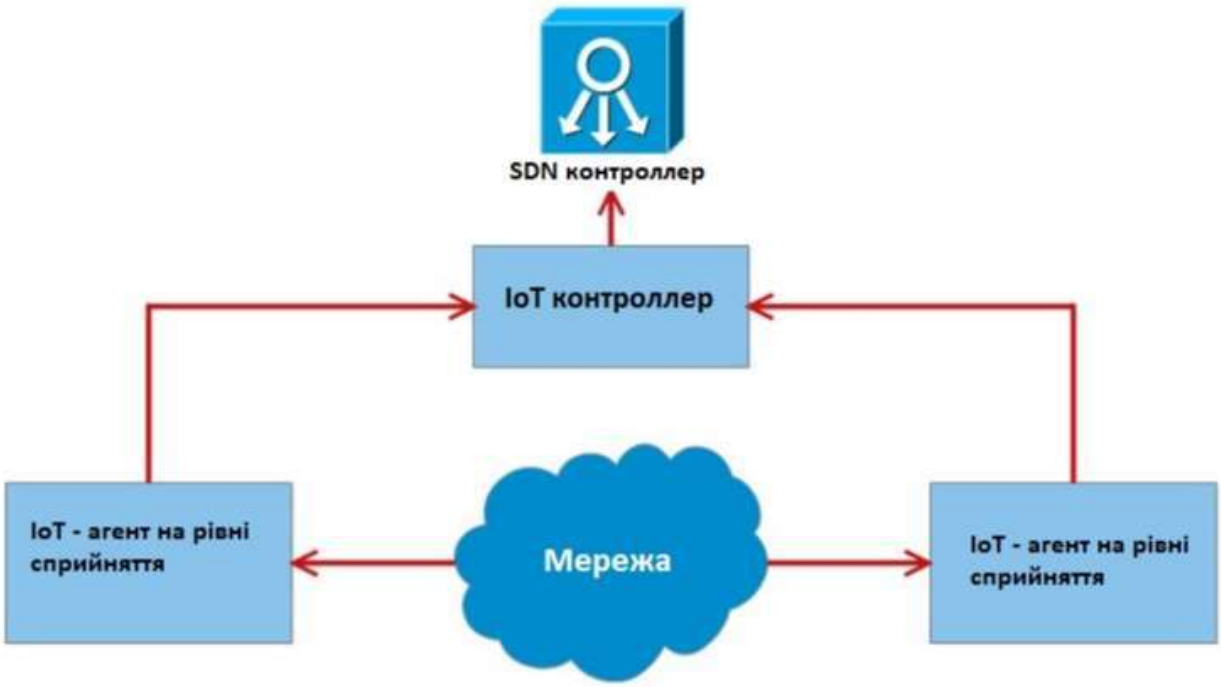


Рисунок 2.2 - Об'єднання SDN з IoT як рішення для безпеки на рівні сприйняття

Вимоги до безпеки рівня сприйняття: перш за все, система IoT повинна бути фізично захищена від фізичного доступу до зловмисника. Аутентифікація вузлів також необхідна для запобігання незаконного доступу до системи. Конфіденційність даних, що передаються між вузлами, дуже важлива, тому для безпечної передачі даних між вузлами повинні бути розроблені полегшені криптографічні алгоритми. Управління ключами також є проблемою, яка повинна бути вирішена в контексті IoT.

2.1.2 Безпека на мережевому рівні

Ядро мережі має достатні заходи безпеки, але деякі проблеми все ще існують. Традиційні проблеми безпеки можуть вплинути на цілісність і конфіденційність даних. У Таблиці 2.1 наведено порівняння різних комунікаційних технологій, що застосовуються в IoT у контексті переваг та недоліків безпеки.

Таблиця 2.1 - Порівняння різних комунікаційних технологій, що застосовуються в IoT з перевагами використання та недоліками безпеки.

Мережева техноголія	Методи безпеки	Сфера застосування	Переваги використання	Недоліки
ZigBee	Шифрування, цілісність	Дім та промисловість	Низьке споживання, низька вартість	Фіксований ключ
Bluetooth	Шифрування, аутентифікація	КПК, мобільні телефони та ноутбуки	Заміна кабелю, низька вартість	Несанкціонований доступ, відправка незапитуваних повідомлень
RFID	Шифрування (AES, DES)	Охорона здоров'я	Захоплення даних без дублювання	Немає авторизації
WSN	Ключ, шифрування, аутентифікація	Будівлі та охорона здоров'я	Низька вартість, потужність та стійкість	DOS-атака
Wi-Fi	Аутентифікація, Авторизація	ПК, телефони та камери	Швидкість, зручність	Прослуховування

Багато типів мережевих атак, такі як атака підслуховування, DoS-атака, атака "Людина на середині" (man in the middle) і вірусне вторгнення, все ще впливають на мережевий рівень:

- Проблеми неоднорідності: рівень сприйняття IoT являє собою поєднання багатьох різнорідних технологій. Мережа доступу має методи множинного доступу, ця неоднорідність робить безпеку і взаємодію більш складною задачею [15];
- Проблеми перевантаження мережі: великий обсяг даних датчика разом з комунікаційними витратами, викликаними автентифікацією великої кількості пристроїв, може викликати перевантаження мережі [15]. Ця проблема повинна

бути вирішена за допомогою можливого механізму аутентифікації пристрою і компетентних транспортних протоколів;

- RFID-втручання: це атака на мережевому рівні, при якій радіочастотні сигнали, які використовуються RFID, пошкоджуються шумовими сигналами, що призводить до відмови в обслуговуванні [17];
- Уповільнення вузла в WSN: це тип атаки, аналогічний радіочастотним перешкодам, як обговорювалося вище для RFID. У цій атаці зловмисник втручається в радіочастоту бездротових сенсорних мереж і відмовляє в обслуговуванні WSN [18]. Це також тип відмови в обслуговуванні;
- Атака підслуховування: це прослуховування трафіку в бездротовому середовищі від WSN, RFID або Bluetooth [19] в зв'язку з бездротовою природою рівня пристроїв в IoT. Кожен тип атаки починається зі збору інформації через прослуховування за допомогою деяких інструментів прослуховування, таких як прослуховування пакетів [20].
- Відмова в обслуговуванні (DoS): зловмисник перевантажує мережу трафіком, що перевищує її пропускну здатність, і, таким чином, мережа недоступна для надання корисних послуг законним користувачам;
- RFID-spoofing: атакуючий підробляє сигнали RFID і зчитує мітку RFID, потім він відправляє підроблені дані з оригінальною міткою RFID і отримує повний доступ до системи;
- Маршрутизаційні атаки: зловмисник може змінювати інформацію про маршрутизацію і поширювати її в мережі для створення циклів маршрутизації, реклами помилкових маршрутів, відправки повідомлень про помилки або скидання мережевого трафіку;
- Атака “Sybil”: в атаці Sybil один шкідливий вузол претендує на ідентичність багатьох вузлів і видає себе за ці вузли. Цей вузол може заподіяти багато шкоди, як, наприклад, поширення неправдивої інформації про маршрутизацію, а також може порушити виборчий процес WSN;

Вимоги до безпеки мережевого рівня: хоча існуюча базова мережева безпека досить сформована, все ж існують деякі проблеми безпеки, які є більш

шкідливими в контексті IoT, такі як відмова в обслуговуванні (DoS) і розподілена відмова в обслуговуванні (DDos) і вони повинні бути попереджені на цьому рівні. Комунікаційні протоколи повинні бути дуже добре сформовані для вирішення проблеми атак на маршрутизацію, проблеми перевантаження і підміни безпеки.

2.1.3 Рівень підтримки

Якщо згадати класичну архітектуру, яка описана у Розділі 1, то мережевий рівень знаходиться між рівнем обробки та проміжним рівнем. Якщо розглядати проміжний рівень та рівень обробки у контексті безпеки, то їх можна прирівняти. Адже вони обидва зберігають, аналізують та обробляють величезну кількість даних. Вони можуть керувати і надавати різноманітний набір послуг нижчим верствам. Тут використовується багато технологій, таких як бази даних, хмарні обчислення та великі модулі обробки даних. У контексті інформаційної безпеки ці два рівня можна об’єднати в один: рівень підтримки.

Безпека рівня підтримки не залежить від інших рівнів, а безпека хмарних обчислень є великою областю безпеки. Cloud Security Alliance (CSA) встановлює безліч стандартів безпеки для хмар. А також розробка механізму безперервного хмарного аудиту, такого як протокол захисту безпеки (SCAP) і надання надійних результатів за допомогою довірених обчислень (TCG). Цей рівень містить дані і додатки користувача IoT, тому дані і додатки повинні бути захищені від порушень безпеки. Деякі проблеми безпеки на цьому рівні:

- Безпека даних: для збереження конфіденційності та захисту даних у хмарі вони повинні бути захищені від взлому. Це можна зробити за допомогою інструментів для виявлення витоку даних із хмари, інструментів запобігання втрат даних, моніторингу активності файлів та баз даних. Дисперсія та фрагментація даних також можуть використовуватися для захисту даних у хмарі;

- Взаємодія та портативність: інтероперабельність та портативність серед постачальників хмарних технологій є головною проблемою сьогодні. Різні постачальники використовують різні власні стандарти, створюючи проблеми для користувачів, які хочуть перейти з однієї хмари на іншу. Ця неоднорідність також створює поблеми з безпекою;

- Забезпечення безперебійного функціонування і післяаварійного відновлення: постачальники хмарних послуг повинні надавати послуги в разі стихійних лих, таких як повені, пожежі і землетруси. Для забезпечення безперервності діяльності "хмари" фізичне місце розташування має бути відповідним, щоб на нього як мінімум не впливали такі лиха. Це повинно бути в підході груп швидкого реагування. Хмари повинні також мати деякі плани резервного копіювання даних;

- Аудит «хмари»: Cloud Security Alliance встановлює багато стандартів для постачальників хмарних технологій, необхідний постійний аудит, щоб перевірити відповідність цих стандартів безпеки для формування довіри користувачів;

- Безпека віртуалізації: різні постачальники "хмарних" обчислень використовували різні методи віртуалізації. Безпека віртуалізації важлива. Зв'язок віртуальних машин деякий час обходить контроль мережевої безпеки. Потрібна безпечна мобілізація віртуальної машини, оскільки вона може стати перешкодою в "хмарному" аудиті.

Вимоги до безпеки на рівні підтримки: користувацькі екземпляри даних і додатків Інтернету речей знаходяться на хмарних і туманних вузлах. Там безпека і конфіденційність не повинні порушуватися в хмарі. CSA вже встановив багато стандартів безпеки, законів і правил для хмарної безпеки. Відповідність цим стандартам безпеки повинна постійно контролюватися, і системи IoT повинні використовувати тільки ті хмари, які відповідають стандартам безпеки CSA. Крім того, цей простий і онлайновий механізм хмарного аудиту необхідний користувачам для перевірки своїх постачальників хмарних обчислень для підвищення довіри користувачів.

2.1.4 Безпека на прикладному рівні

Різні програми на прикладному рівні мають різні вимоги до безпеки. В даний час не існує стандарту для побудови додатків IoT. Однак обмін даними є однією з характеристик прикладного рівня IoT. Спільне використання даних стикається з проблемами конфіденційності даних і контролю доступу. Деякі з загальних питань безпеки на прикладному рівні є такими:

- Доступ до даних і аутентифікація: у додатку може бути багато користувачів, і різні користувачі можуть мати різні привілеї доступу. Належний механізм аутентифікації і контролю доступу потрібний на рівні додатку [15];
- Фішингові атаки: зловмисник використовує заражені електронні листи або веб-посилання для крадіжки законних облікових даних користувача і отримання доступу з використанням цих облікових даних;
- Шкідливі X-скрипти: зловмисник може відправити активний X-скрипт користувачеві IoT через Інтернет і змусити користувача IoT запустити цей скрипт, тим самим ставлячи під загрозу всю систему;
- Атака за допомогою шкідливих програм: зловмисники можуть атакувати додатки, використовуючи шкідливі програми, і можуть красти дані або викликати відмову в обслуговуванні. Троянські коні, черв'яки і віруси - ось деякі з небезпечних шкідливих програм, що використовуються зловмисниками для експлуатації системи.

Вимоги до безпеки на прикладному рівні: щоб впоратися з безпекою на прикладному рівні, необхідний потужний механізм аутентифікації і контролю доступу. Крім того, важливо навчити користувачів використовувати надійний пароль. Для захисту від шкідливих програм необхідні потужні антивірусні програми.

На основі аналізу проблем безпеки на кожному рівні IoT-мереж, була складена таблиця, яка показує коротку характеристику та рівень загроз на

кожному рівні архітектури IoT (табл. 2.2) та контрзаходи на кожному рівні (табл. 2.3).

Таблиця 2.2 Рівень загроз в архітектурі IoT

Тип атаки	Рівень сприйняття	Мережевий рівень	Рівень підтримки	Прикладний рівень	Вплив
Втручання у вузол	✓	✗	✗	✗	Високий
Підроблення вузла	✓	✗	✗	✗	Високий
Атака на боковий канал	✓	✗	✗	✗	Середній
Фізичні ушкодження	✓	✗	✗	✗	Середній
Введення шкідливого коду	✓	✗	✗	✓	Високий
Захист даних датчиків	✓	✗	✗	✗	Середній
Масова аутентифікація вузлів	✓	✓	✓	✗	Високий
Проблеми неоднорідності	✗	✓	✓	✗	Високий
Проблеми перевантаження мережі	✗	✓	✓	✗	Середній
RFID-втручання	✗	✓	✓	✗	Низький
Уповільнення вузла в WSN	✗	✓	✓	✗	Низький
Атака підслуховування	✗	✓	✓	✗	Низький
Відмова в обслуговуванні (DoS)	✗	✓	✓	✗	Високий
RFID-spoofing	✗	✓	✓	✗	Високий

Продовження Таблиці 2.2

Маршрутизаційні атаки	✗	✓	✓	✗	Високий
Атака "Sybil"	✗	✓	✓	✗	Високий
Захист даних	✗	✗	✗	✗	Високий
Взаємодія та портативність	✗	✓	✓	✗	Середній
Безперебійне функціонування та післяаварійне відновлення	✗	✗	✗	✗	Середній
Аудит хмарного середовища	✗	✗	✗	✗	Середній
Орендна безпека	✗	✗	✗	✗	Високий

Безпека віртуалізації	✗	✗	✗	✗	Середній
Доступ до даних та аутентифікація	✗	✗	✗	✓	Високий
Фішингові атаки	✗	✗	✗	✓	Середній
X-скрипти	✗	✗	✗	✓	Високий
Шкідливі програми	✗	✗	✗	✓	Високий

Таблиця 2.3 – Контрзаходи безпеки на рівнях архітектури IoT

Архітектурний рівень IoT	Контрзаходи
Рівень сприйняття	Фізична безпека поблизу вузлів, Потреба в легкому алгоритмі шифрування обмежених вузлів, конфіденційність даних датчиків, ефективні механізми аутентифікації та контролю доступу для пристроїв, механізми атак Anti DoS
Мережевий рівень	Протоколи захищеного зв'язку від атак повторного відтворення, атак маршрутизації, атак з перешкодами; обробка перевантажень і захист від DDoS-атак в протоколах зв'язку.
Рівень підтримки	Необхідність безперервного хмарного аудиту, впровадження стандартів Cloud Security Alliance, технологій безпечної віртуалізації, поділу клієнтів, шифрування сховища для користувачів, конфіденційності та цілісності даних.
Прикладний рівень	Безпечний код додатку, навчання користувачів використанню складних паролів, механізми контролю доступу, узгодження ключів, моніторинг журналів, інструменти моніторингу файлів і баз даних, захист від шкідливих програм.

2.2 Механізми захисту безпеки в IoT

Безпека є основною вимогою будь-якого користувача цифрових послуг. Користувач Інтернету не буде ділитися своїми конфіденційними і важливими даними в мережі, якщо мережа не є довіреною.

Кожен розроблювальний IoT-пристрій складається з різних датчиків, протоколів підключення і т.д., які піддаються впливу різних загроз атак. Наприклад, неавторизований доступ, аналіз або зміна трафіку, підміна сертифікатів і оновлення прошивок на пристроях. Забезпечення механізму безпеки пристроїв повинно розглядатися в кожному створюваному IoT проект. З метою захисту екосистеми IoT необхідно задіяти механізми безпеки на різних

рівнях розробки та впровадження IoT-пристроїв. Механізми забезпечення безпеки наведені в Таблиці 2.4.

Таблиця 2.4 – Механізми забезпечення безпеки в IoT

Рівень архітектури IoT	Механізми захисту
Рівень сприйняття	Перевірка достовірності
	Аутентифікація та контроль доступу
	Служба конфіденційності
	Механізм цифрового підпису
	Механізм PKI
Мережевий рівень	Служба доступності
	Антивірусна служба
	Шифрування даних
	Аутентифікація та контроль доступу
Рівень підтримки	Фільтрація маршрутизаторів
	Служба доступності
	Аутентифікація та контроль доступу
	Журнал аудиту
	Механізм IDS
	Моніторинг подій
Прикладний рівень	Аутентифікація та контроль доступу
	Служба доступності
	Антифішингова служба

Продовження таблиці 2.4

	Антивірусна служба
	Механізм системи виявлення порушень
	Механізм фільтрації спаму

На основі розглянутих доступних механізмів захисту в IoT на Рисунку 2.3 зображено діаграму, на якій видно, що найпоширеніший механізм, який застосовується на кожному рівні IoT, це аутентифікація та контроль доступу.

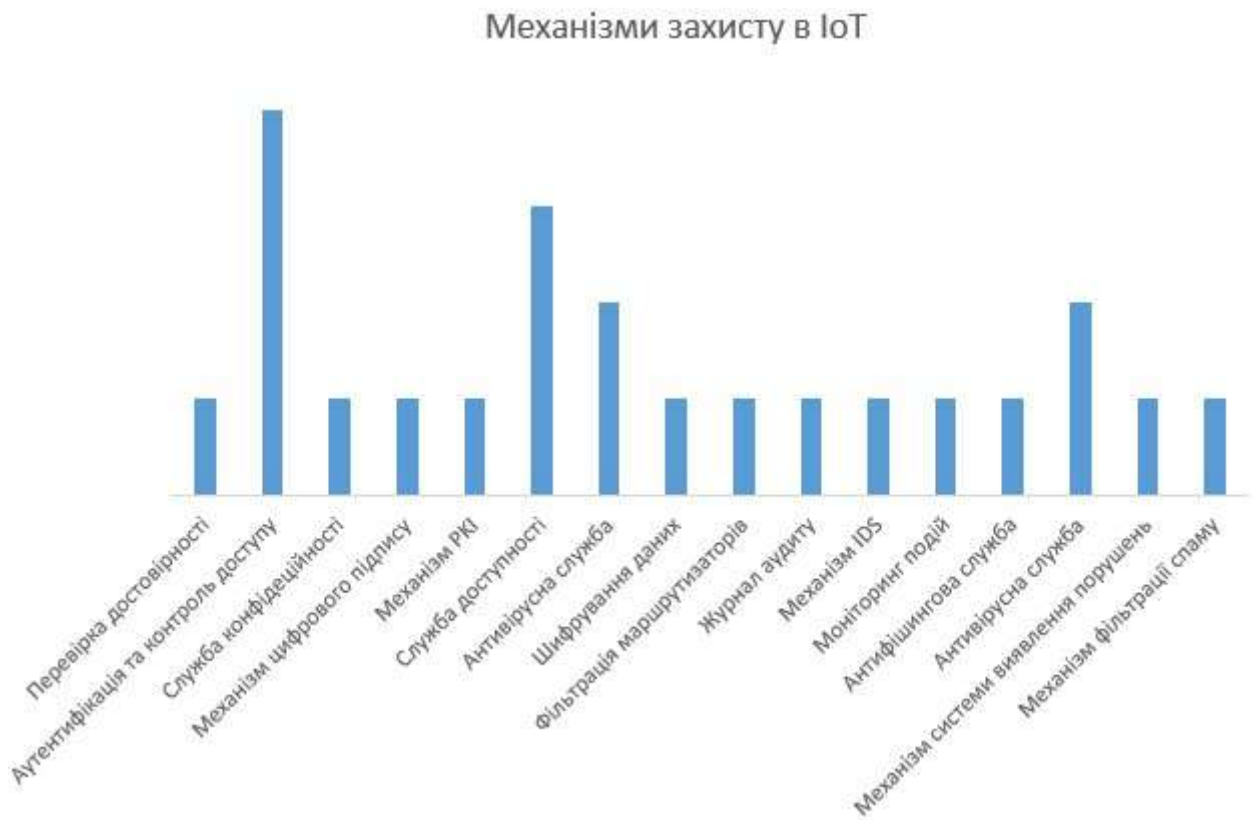


Рисунок 2.3 – Найпоширеніші механізми захисту в IoT

Тому далі буде докладно розглянуто важливість та методи аутентифікації та контролю доступу.

2.3 Аутентифікація та контроль доступу в IoT

Безпека Інтернету речей, яка сьогодні є актуальною темою, містить безліч проблем безпеки і конфіденційності. Через величезну кількість IoT-пристроїв і міжмережевої взаємодії IoT застарілі методи аутентифікації і авторизації для нього недоцільні. Пристрої повинні перевіряти достовірність один одного до обміну будь-якою інформацією між ними (зв'язок M2M), що є проблемою через величезну кількість пристроїв.

IoT фокусується на режимі зв'язку між машинами (M2M) [15]. Для таких вузлів зв'язку аутентифікація дуже важлива для забезпечення безпеки і конфіденційності. Коли два або більше вузла обмінюються даними один з

одним для досягнення спільної мети, вони повинні спочатку підтвердити справжність один одного, щоб заблокувати атаку помилкового вузла. Однак не існує ефективного механізму аутентифікації для величезної кількості пристроїв IoT.

Далі будуть наведені деякі методи, пов'язані з аутентифікацією пристрою і контролем доступу в IoT.

2.3.1 Модель управління доступом для розподіленого середовища IoT

Така модель підтримує загальний режим з використанням одного токена і гарантує наскрізну безпеку з використанням IPsec (Рисунок 2.4). Запитуюча сторона може використовувати один токен для групового доступу (група пристроїв, що пропонують загальні послуги) для зв'язку з будь-яким пристроєм в групі. Мережевий префікс унікального локального ідентифікатора (ULA) використовується в якості ідентифікатора групи доступу. Кожен пристрій в групі ідентифікується ULA. У токені групового доступу запитуюча сторона поміщає свій ULA і мережевий префікс групи доступу. Отже, пристрої в групі можуть перевіряти токен, використовуючи свій ULA і префікс в токені. Він також може забезпечувати управління доступом на основі запиту ULA в токені.

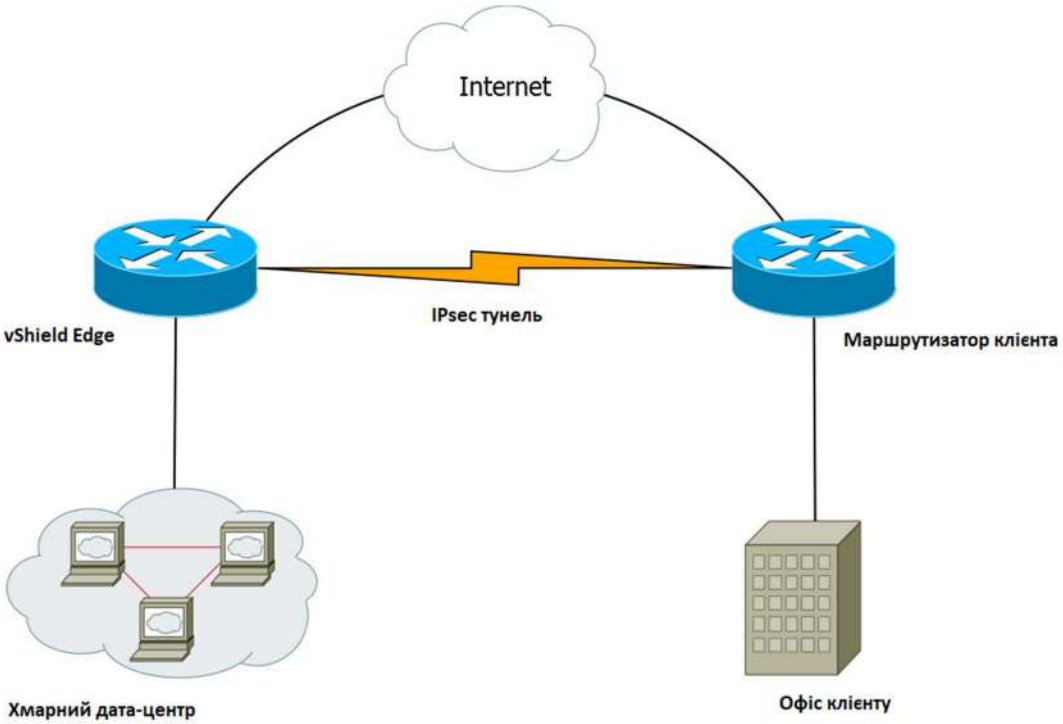


Рисунок 2.4 – Принцип роботи IPsec

2.3.2 Структура управління доступом для середовища IoT «SmartOrBAC»

Структура управління доступом для середовища IoT «SmartOrBAC» заснована на моделі OrBAC (Рисунок 2.5) (управління доступом на основі організації). Ця модель використовує веб-сервіси для забезпечення дотримання політик безпеки. Управління доступом на основі організації (OrBAC) має деякі обмеження, наприклад, воно краще працює в централізованій системі, воно не зачіпає співпрацю між організаціями і суборганізаціямі, а OrBAC не переводить політику безпеки в механізм управління доступом. Тому для усунення цих обмежень пропонується SmartOrBAC, який є розширенням OrBAC. SmartOrBAC використовує веб-сервіси для забезпечення безпечного співробітництва між різними організаціями. Взаємодія між організаціями визначається угодою між організаціями. Організації спільно визначають правила доступу відповідно до формату OrBAC. У SmartOrBAC контракт не укладено апріорі, але він може бути виконаний на льоту спонтанно і динамічно. SmartOrBAC забезпечує ефективне управління доступом для об'єктів спільної

роботи з сценаріями з низьким енергоспоживанням і енергоспоживанням, такими як IoT.

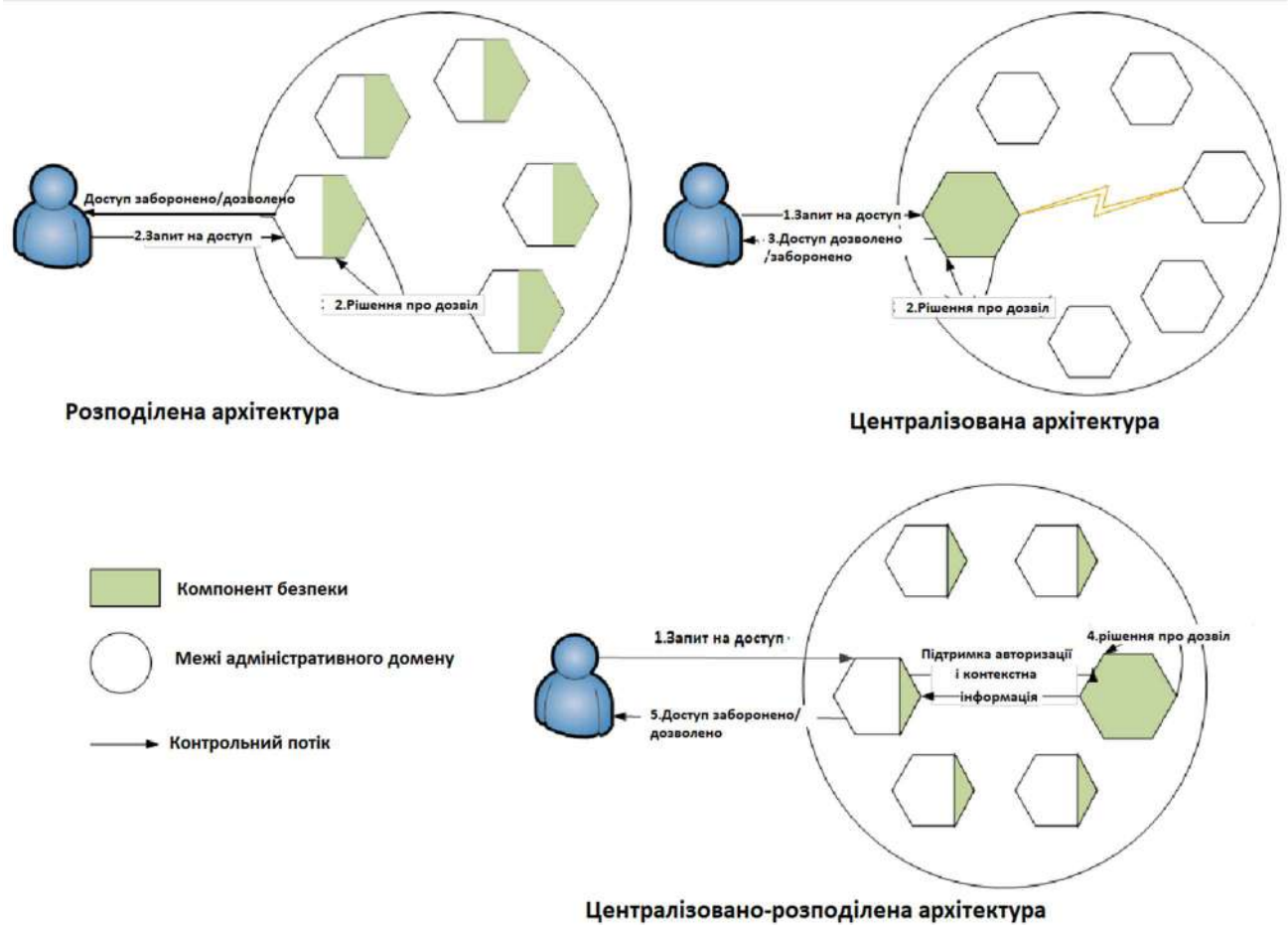


Рисунок 2.5 – Архітектура SmartOrBAC

2.3.3 Структура управління доступом рівня обслуговування для пристроїв з обмеженою потужністю

Структура управління доступом рівня обслуговування для пристроїв з обмеженою потужністю дозволяє за кожну послугу детально контролювати доступ. Вона використовує кращі протоколи обмеженого застосування (CoAP) для створення платформи з низьким енергоспоживанням для управління

доступом і аспектів аутентифікації (Рисунок 2.6). Клієнт CoAP отримує тікет від сервера CoAP і використовує цей тікет в кожному майбутньому запиті CoAP. Є два кроки для аутентифікації і для контролю доступу. Користувач спочатку аутентифікується на основі облікових даних, таких як загальний ключ, пароль або інший валідатор. При успішній аутентифікації CoAP NAS інформується про користувачів і їх дозволах, часу відсутності заявки, групі і т. п. CoAP-NAS відправляє користувачу тікет для майбутніх запитів. На етапі контролю доступу сервер відповість правильним повідомленням тільки в тому випадку, якщо повідомлення запиту має дійсний тікет, в іншому випадку буде згенеровано повідомлення про помилку.

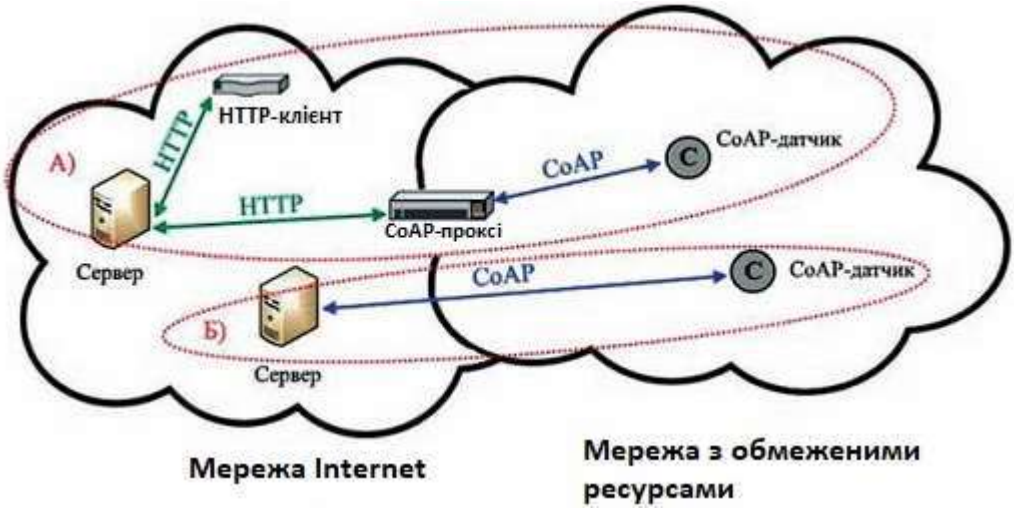


Рисунок 2.6 – Взаємодія з протоколом CoAP

2.3.4 Механізм захисту IoT з використанням цифрових сертифікатів із захистом транспортного рівня дейтаграми (DTLS)

Для безпечного зв'язку в IoT аутентифікація виконується за допомогою цифрових сертифікатів, наданих центром сертифікації, які роблять аутентифікацію більш надійною і замінюють механізм попереднього ключа в DTLS. Клієнт/сервер аутентифікується шляхом перевірки підпису за допомогою цих кроків:

1. Клієнт відправляє запит на сервер;

2. Сервер відправляє свій сертифікат клієнту;

3. Клієнт перевіряє сертифікат, розшифровуючи його за допомогою відкритого ключа сервера;

4. Після перевірки клієнт відправляє свій власний сертифікат на сервер;

5. Сервер здійснює перевірку за допомогою тієї ж процедури, а потім вони можуть почати зв'язок.

2.3.5 Метод для забезпечення надійної безпеки в невеликих мережах (системи розумного будинку)

Система заснована на фреймворку AllJoyn. Система працює в мережі Wi-Fi, і є вузол шлюзу Wi-Fi, який відповідає за початкову конфігурацію системи, аутентифікацію пристроїв IoT і надає користувачеві засіб управління системою за допомогою мобільного пристрою за допомогою додатка Android. Процес аутентифікації складається з двох етапів: з мобільного пристрою на пристрій IoT (користувач завантажує ідентифікаційні дані та попередній загальний ключ і після цього домашні облікові дані взаємної аутентифікації передаються на пристрій IoT) і шлюз на пристрій IoT (пристрій IoT підключається до шлюзу, а шлюз аутентифікує його за допомогою інформації, відправленої мобільним користувацьким пристроєм). Після цього відбувається зашифрований зв'язок.

2.3.6. Полегшений протокол аутентифікації

Такий метод працює за рахунок посилення вихідної бази безпеки системи RFID на IoT. В існуючому протоколі аутентифікація RFID виконується без шифрування, що є недоліком безпеки. Для вирішення цієї проблеми

пропонується легкий криптографічний протокол, заснований на методі XOR, за допомогою якого для аутентифікації використовуються зашифровані паролі.

2.4 Висновки за розділом 2

Безпека Інтернету речей - активна тема для дослідження. У цьому розділі розглядаються основні проблеми безпеки на кожному рівні архітектури IoT. Також в цьому розділі представляються короткі контрзаходи для різних проблем безпеки для захисту систем IoT, обговорюються проблеми, пов'язані із застарілими рішеннями безпеки в IoT, досліджується механізм аутентифікації і контролю доступу в IoT. Застарілий механізм аутентифікації не підходить для пристроїв IoT, оскільки ці пристрої обмежені в ресурсах і їх велика кількість. Тому для аутентифікації обмежених пристроїв в M2M-зв'язку потрібен новий механізм.

3 АРХІТЕКТУРА ВИПРОБУВАЛЬНОГО СТЕНДУ ДЛЯ АНАЛІЗУ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ В ІНТЕРНЕТІ РЕЧЕЙ

Пропонована архітектура випробувального стенду, представлена в цьому розділі, заснована на потребах кінцевих користувачів, пріоритетних сценаріях ризику, законах про регулювання, на кращих практиках і стандартах, а також на архітектурі екосистеми Інтернету речей, включаючи опис модулів випробувального стенда.

3.1 Можливості випробувального стенду та вимоги

Необхідні можливості та вимоги для випробувального стенду для аналізу інформаційної безпеки в IoT можуть бути класифіковані і сформульовані на різних рівнях в такий спосіб (Рисунок 3.1) і будуть описані далі.

Рисунок 3.1 – Можливості та вимоги випробувального стенду

3.1.1 Ініціалізація та виявлення

Використовуючи різні симулятори, стимулятори і будь-які інші необхідні інструменти, на випробувальному стенді слід імітувати реальні умови для тестування пристроїв IoT в різних контекстах. Після ініціалізації і активації пристрою IoT наступною вимогою є виявлення пристрою IoT, присутнього в середовищі випробувального стенду. У процесі виявлення повинен бути створений файл журналу, що складається з операційної системи пристрою IoT (OS), запущених процесів, виконуваних дій і т. п. Ця інформація буде використовуватися для будь-якого подальшого виявлення аномалій.

3.1.2 Перевірка безпеки

Випробувальний стенд з аналізу безпеки в IoT-пристроях повинен підтримувати ряд перевірок безпеки, кожен з яких орієнтований на певний аспект безпеки. Випробувальний стенд повинен виявляти різні уразливості, до яких можуть бути схильні до пристрою IoT, і забезпечувати аналіз цих вразливостей. Відповідно, випробувальний стенд з аналізу безпеки повинен враховувати деякі з вразливостей включаючи: проникнення, непрацюючу аутентифікацію і управління сеансами, міжсайтовий скриптинг (XSS), неправильне налаштування безпеки, вразливість даних, відсутність контролю доступу на функціональному рівні і використання компонентів з відомими вразливостями. Крім того, на випробувальному стенді повинні підтримуватися шаблони тестів безпеки і сценарії. Випробувальний стенд повинен бути здатний виконувати автоматичні тести на основі конкретних вимог (наприклад, вилучати всі тести, які мають відношення до датчика акселерометра) або типу пристрою (наприклад, всі тести, які відносяться до IP-камерам). Крім того, на випробувальному стенді повинен бути передбачений критерій успіху кожної перевірки безпеки (наприклад, шкала від 1 (проходження) до 5 (збій)), який може грунтуватися на заздалегідь визначеному пороговому значенні, заздалегідь наданому системним оператором.

3.1.3 Реєстрація та аналіз

Після виконання ряду кроків, пов'язаних з функціональними вимогами, випробувальний стенд повинен бути здатний реєструвати випробування. Система збирає різні дані під час виконання перевірки, включаючи інформацію про трафік мережі (наприклад, про роботу Wi-Fi, Bluetooth і ZigBee), інформацію про внутрішній стан пристрою IoT (наприклад, використання CPU, споживання пам'яті та активність файлової системи) і т. п. Ця інформація повинна зберігатися у вигляді файлу журналу для подальшого аналізу. Крім того, система випробувального стенду повинна підтримувати інтелектуальний аналіз.

3.1.4 Зручність використання

Зручність використання забезпечує простоту використання випробувального стенду з мінімальними зусиллями з боку користувача. Випробувальний стенд для аналізу безпеки повинен бути простим в експлуатації і використанні, з легко визначеними тестами, простою вводимою конфігурацією і легкою інтерпретацією вихідних даних.

3.1.5 Зв’язок з безпекою

1. Надійність: здатність випробувального стенду для аналізу безпеки IoT-пристроїв виконувати свої необхідні функції в зазначених умовах протягом певного періоду часу;

2. Реагування: здатність випробувального стенду виявляти і згодом запобігати активації шкідливих додатків на пристрої IoT (якщо він був заражений);

3. Безпека: здатність випробувального стенду забезпечувати авторизований доступ до системи для захисту цілісності стенду від випадкового або зловмисного пошкодження;

4. Звітування (включаючи безвідмовність): можливість випробувального стенду вести записи аудиту для підтримки незалежної перевірки доступу до ресурсів / використанням випробувального стенду.

3.1.6 Адаптивність

Випробувальний стенд безпеки повинен бути в змозі адаптуватися відповідно до нових концепцій предметної області і підтримувати різні типи зв'язку.

1. Масштабованість: здатність випробувального стенду збільшувати загальну пропускну здатність при збільшеною навантаженні, коли ресурси (зазвичай програмні і апаратні) додаються до випробувального стенду.
2. Продуктивність: здатність випробувального стенду добре працювати в різних умовах, наприклад, продуктивність по відношенню до часу і введення даних користувачем.
3. Гнучкість: можливості зміни випробувального стенду після установки. Це включає в себе адаптованість, стійкість і налаштовуваність.

3.2. Архітектура системи

Архітектура випробувального стенду для аналізу безпеки IoT-пристроїв, показана на Рисунку 3.1, являє собою засновану на рівнях архітектури Інтернету речей платформу з модульною структурою. Це означає, що будь-який тип пристрою IoT може бути протестований в запропонованій платформі тестування безпеки, включаючи інтелектуальні пристрої, інтелектуальні міські пристрої, інтелектуальні носимі пристрої та багато іншого.

Крім того, для виконання перевірки безпеки в різних контекстах в середовищі випробувального стенду можуть бути розгорнуті будь-які відповідні імітатори і/або стимулятори, а також інструменти вимірювання або аналізу, що використовуються для збору і аналізу результатів випробувань.

Як модульна система, випробувальний стенд також об'єднує різні механізми аналізу у вигляді плагінів, які використовуються для проведення розширеного тестування безпеки (в основному це механізми, засновані на розроблених алгоритмах машинного навчання). Далі наведено детальний опис модулів, які складають функціональну модель, і взаємодії між цими модулями як повної системи тестування безпеки.

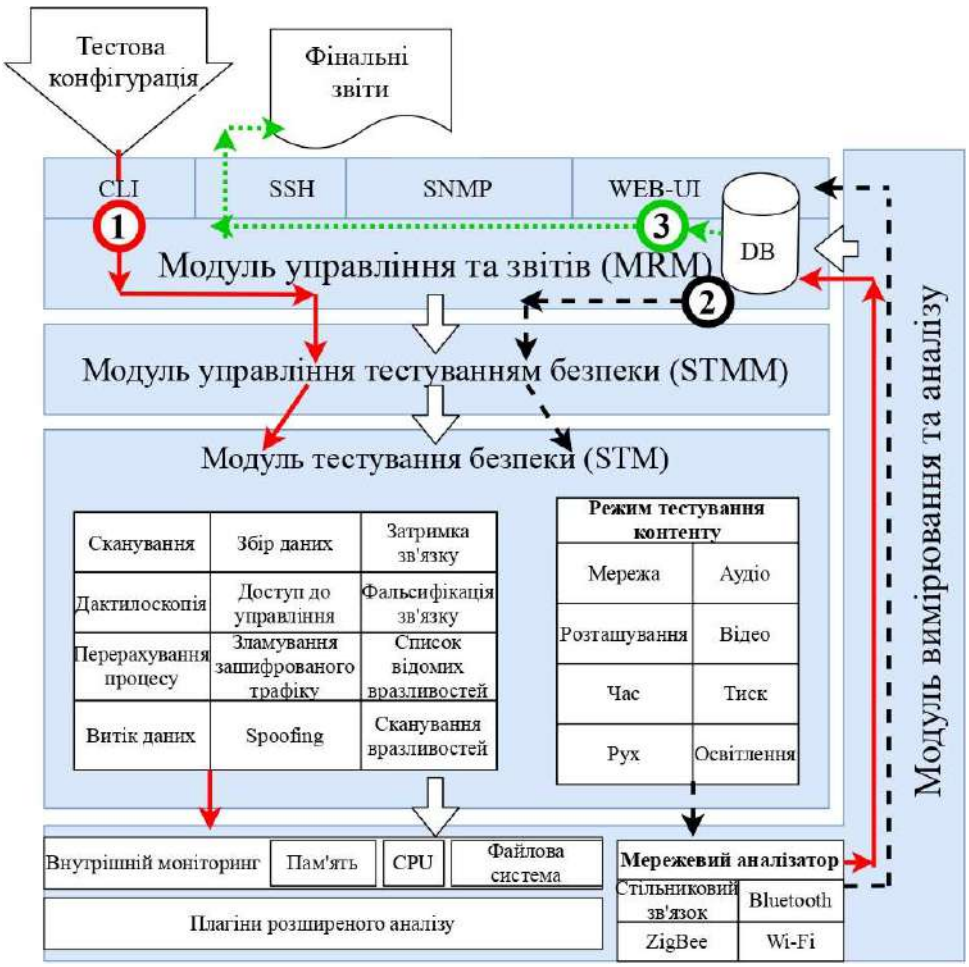


Рисунок 3.2 - Структура випробувального стенду для аналізу безпеки в IoT – Абстрактна модель функціональної архітектури

3.2.1 Модуль управління та звітів (MRM)

Цей модуль відповідає за ряд дій з управління та контролю, включаючи запуск/ініціалізацію процедури тестування, реєстрацію нових пристроїв, симуляторів / стимуляторів, тести безпеки, інструменти вимірювання та аналізу на випробувальному стенді і генерацію остаточних звітів по завершенню перевірки. Користувач взаємодіє з випробувальним стендом через цей модуль, використовуючи один з інтерфейсів зв'язку (інтерфейс командного рядка / SSH / простий протокол управління мережею (SNMP) / веб-інтерфейс користувача (WEB-UI)) для того, щоб почати тестування, а також для отримання остаточних звітів. Відповідно, цей модуль взаємодіє з модулем управління тестуванням

безпеки (STMM) і модулем вимірювання і аналізу (MAM) відповідно. MRM містить компонент системної бази даних, де зберігається вся необхідна інформація про тестований пристрій (включаючи OS, можливості підключення, можливості датчиків, розширені функції і т. п.), а також інформацію про сам тест (включаючи файли конфігурації, знімки системи і результати тесту).

3.2.2. Модуль управління тестуванням безпеки (STMM)

Цей модуль відповідає за фактичну послідовність тестування, виконувану випробувальним стендом. Відповідно, він взаємодіє з модулем тестування безпеки (STM) для виконання необхідного набору тестів в правильному порядку і режимі з використанням заданих параметрів конфігурацій, наданих користувачем (на основі файлу конфігурації, завантаженого в MRM).

3.2.3. Модуль тестування безпеки (STM)

Цей модуль виконує стандартне тестування безпеки на основі оцінки вразливості і методології тестування на проникнення, щоб оцінити рівень безпеки IoT-пристрою, що перевіряється. STM - це операційний модуль, який виконує набір тестів безпеки у вигляді плагінів, кожен з яких виконує певне завдання в процесі тестування. Цей модуль також підтримує контекстний режим тестування, де він генерує різні стимули навколишнього середовища для кожного датчика або пристрою. Це означає, що в цьому режимі роботи STM моделює різні тригери середовища і запускає тести безпеки для моделювання різних контекстів і робочих середовищ для протестованих пристроїв IoT. Це виходить з використанням списку масивів симуляторів, таких як симулятор GPS або симулятор локалізації Wi-Fi (для атак з урахуванням місця розташування і на основі геолокації), симулятора часу (з використанням

симульованої мережі, симулятора GPS або локального NTP-сервера), симулятор руху (наприклад, з використанням роботів) і т. п. Список підтримуваних симуляторів див. в Таблиці 3.2. STM взаємодіє з модулем вимірювань і аналізу (MAM) для моніторингу виконаного тесту і аналізу результатів тесту.

Таблиця 3.1. – Симулятори, які підтримуються випробувальним стендом

Симулятор	Опис
Мережа	На випробувальному стенді можуть використовуватися симулятори мережі для моделювання різних мережевих середовищ, таких як Wi-Fi, Bluetooth, ZigBee і т.д., для підтримки різних мережевих підключень на випробувальному стенді.
Розташування	Випробувальний стенд може імітувати різні місця розташування і траєкторії з використанням генерувального пристрою GPS, щоб перевірити поведінку тестованого пристрою IoT в різних місцях / траєкторіях.
Час	Тестовий стенд імітує різні дні тижня і час доби, використовуючи або генеруючий пристрій GPS, внутрішній NTP-сервер або внутрішню стільникову мережу, щоб протестувати поведінку тестованого пристрою IoT в різний час.
Рух	Випробувальний тест імітує різні рухи за допомогою роботів чи тестерів для того, щоб перевірити поведінку тестуваного пристрою IoT під час виконання різних рухів.
Освітлення	Випробувальний стенд може імітувати різні рівні освітлення, щоб перевірити поведінку тестуваного пристрою LoT в різних сценаріях освітлення.
Аудіо	Випробувальний стенд може імітувати аудіо за допомогою голосового симулятора, щоб перевірити поведінку пристрою IoT, що тестується, в різних звукових середовищах.

3.2.4 Модуль вимірювання та аналізу (MAM).

У цьому модулі використовуються різні компоненти вимірювання, збору даних і аналізу. Вимірювальні компоненти включають в себе різні мережеві аналізатори для моніторингу зв'язку, такі як аналізатори Wi-Fi, стільникового зв'язку, Bluetooth і ZigBee, і інструменти моніторингу пристроїв для

вимірювання внутрішнього стану тестувальних пристроїв, включаючи споживання пам'яті, завантаження CPU і зміни файлової системи в IoT-пристроях.

На основі зібраної інформації на випробувальному стенді проводиться розширене тестування безпеки з використанням механізмів, заснованих на розроблених алгоритмах машинного навчання. Компонент аналізу обробляє зібрані дані і оцінює результати відповідно до заздалегідь певному критерію успіху. У деяких випадках критерій успіху не може бути чітко визначено, і, отже, на випробувальному стенді будуть розгорнуті розширені інструменти і механізми аналізу (наприклад, буде використовуватися інструмент виявлення мережевих аномалій для обробки записаного мережевого трафіку перевіреного IoT пристроя для виявлення аномальних подій в системі). В цьому випадку рішення про проходження/відмову буде грунтуватися на заздалегідь заданому пороговому значенні, заздалегідь наданому оператором системи. Потім виявлені аномалії повинні бути досліджені і інтерпретовані оператором системи з використанням спеціальних інструментів дослідження, які є частиною призначеного для користувача інтерфейсу.

3.2.5 Процес перевірки

Процес тестування, показаний на Рисунку 3.1., починається з завантаження файлу конфігурації (користувачем/оператором випробувального стенду) в випробувальний стенд через модуль MRM. На основі завантаженої конфігурації на випробувальному стенді проводиться набір тестів безпеки

(позначений червоною лінією на Рисунку 1) з використанням модуля STM. Результати потім зберігаються в системному компоненті бази даних.

Потім виконується контекстне тестування безпеки з використанням модуля STM (позначеного чорною пунктирною лінією на Рисунку 3.1.) шляхом вибору відповідних імітаторів для тесту. На цьому етапі використовуються різні симулятори для реалістичного моделювання середовища, в якій працюють пристрої IoT, і проводиться один і той же набір тестів безпеки (знову ж таки, на основі попередньо завантаженого файлу конфігурації). Отримані результати потім зберігаються в компоненті бази даних системи. Обидві ці фази тестування контролюються модулем STMM. Слід звернути увагу, що під час виконання процесу тестування використовуються різні інструменти вимірювання та аналізу з використанням модуля MAM для збору відповідної інформації про проведеному тесті (включаючи мережевий трафік, внутрішній стан IoT-пристрою і т. д.).

Нарешті, аналіз виконується модулем MRM на основі результатів, отриманих на обох етапах, і інформації, зібраної в процесі тестування. Остаточні результати всього процесу тестування потім генеруються і відправляються користувачеві / оператору випробувального стенду (позначений зеленою пунктирною лінією на Рисунку 3.1).

3.3 Системна структура та компоненти

Середа випробувального стенду, показана на Рисунку 3.2, включає як програмні, так і апаратні компоненти системи. З точки зору внутрішнього компонента системи програмного забезпечення це включає в себе користувальницький інтерфейс і декілька модулів диспетчера тестування, кожен з яких відповідає за певну задачу. З точки зору зовнішніх компонентів

системи сюди можна включити IoT-пристрої, набір інструментів тестування безпеки, інструменти вимірювання та аналізу, а також набір симуляторів/стимуляторів, які використовуються в випробувальному стенді.

3.3.1 Внутрішні програмні компоненти системи

Внутрішні програмні компоненти системи випробувального стенду для аналізу інформаційної безпеки в пристроях IoT включають в себе користувальницький інтерфейс (GUI / Remote), управління випробувальним стендом, управління тестами, управління елементами і елементи управління зберіганням.

1. Користувальницький інтерфейс - GUI / Remote: компонент призначеного для користувача інтерфейсу використовується для відправки і отримання команд і результатів випробувань на / з випробувального стенду, відповідно. Це може бути виконано локально (наприклад, з використанням графічного інтерфейсу користувача) або віддалено (наприклад, через REST API). SSH і Telnet підключення також підтримуються.

2. Управління випробувального стенду: компонент управління випробувального стенду виступає в якості координатора в системі. Він відповідає за управління робочим процесом між компонентами системи програмного забезпечення випробувального стенду (включаючи базові менеджери: менеджер елементів, менеджер випробувань і т. д.), а також за компоненти апаратного забезпечення системи і користувальницький інтерфейс.

3. Управління тестами: компонент управління тестування відповідає за створення і виконання сценаріїв тестування. Сценарій визначає процес тестування на стенді, включаючи створення і виконання тестів безпеки, кожен з яких складається з набору дій з тестування безпеки. Крім того, диспетчер тестів дозволяє створювати шаблони сценаріїв тестування для подальшого використання.

4. Управління елементами: компонент управління елементів відповідає за надання та вилучення елементів з випробувального стенду. Елемент - це загальний термін, який використовується в випробувальному стенді, який застосовується як до програмного, так і до апаратного забезпечення. Кожен елемент визначається його драйвером. Драйвер - це програмований компонент, який надає можливості елемента як користувачу, так і інших елементів випробувального стенду. Прикладами типів елементів, які використовуються в випробувальному стенді, є: IoT-пристрої, симулятори / стимулятори, інструменти вимірювання та аналізу, а також тести безпеки.

5. Управління зберіганням: компонент управління сховища є сховищем системних елементів. Крім того, він відповідає за реєстрацію різних подій, що відбуваються в системі, до, під час і після проведення тесту (наприклад, реєстрація симулятора, події драйвера, що виконується дію тесту, результати тесту і т. д.)

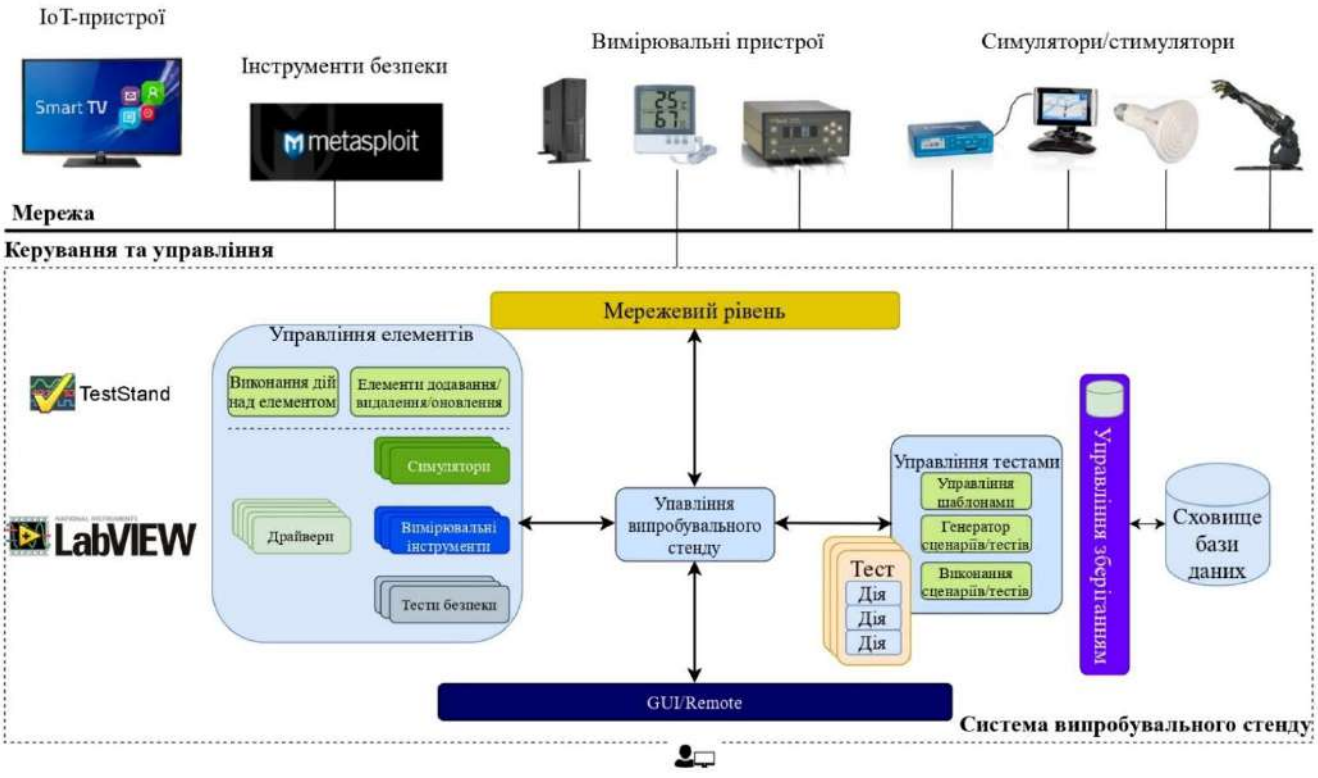


Рисунок 3.3 – Компоненти системи випробувального стенду для IoT

3.3.2. Зовнішні компоненти системи

Компоненти зовнішньої системи включають в себе як апаратні, так і програмні компоненти, в тому числі: IoT-пристрої, набір інструментів безпеки, симулятори та стимулятори зовнішнього середовища, а також різні типи інструментів вимірювання і аналізу.

1. IoT-пристрої: випробувальний стенд безпеки проектується і реалізується для підтримки перевірки широкого спектру пристроїв IoT, включаючи різні категорії, такі як інтелектуальні побутові пристрої, інтелектуальне промислове обладнання, інтелектуальні міські пристрої, носимі пристрої та багато іншого;

2. Інструменти перевірки безпеки: на випробувальному стенді можуть використовуватися різні інструменти тестування безпеки, доступні онлайн, в тому числі сканер безпеки Nmap для виявлення мережі та аудиту безпеки, інструмент Wireshark для аналізу мережевого протоколу, Aircrack-ng для оцінки мереж Wi-Fi, і Metasploit, який використовується для тестування на проникнення. Всі ці інструменти працюють в середовищі тестування на проникнення Kali Linux. Інші засоби безпеки, такі як Nessus, OpenVAS, Cain&Abel і OSSEC, також можуть бути використані в випробувальному стенді;

3. Інструменти вимірювання та аналізу: на випробувальному стенді можуть використовуватися різні типи інструментів вимірювання і аналізу, в тому числі: модулі збору даних, модулі аналізу і оцінки безпеки, модулі аналізу даних і багато іншого. Ці модулі розроблені для розширення можливостей випробувального стенду; наприклад, модель виявлення аномалій може використовуватися в випробувальному стенді для автоматичної ідентифікації та виявлення аномалій в мережевому трафіку IoT-пристроїв.

4. Симулятори та стимулятори: на випробувальному стенді можуть використовуватися різні типи зовнішніх симуляторів і стимуляторів (наприклад, симулятор GPS, який імітує різні місця розташування і траєкторії, симулятори руху, такі як рука робота і т. д.). Використовуючи набір симуляторів, випробувальний стенд реально може генерувати довільні

стимуляції в реальному часі, в ідеалі для всіх датчиків тестованих пристроїв IoT. У Таблиці 3.1 наведено список симуляторів, підтримуваних випробувальним стендом.

3.4 Механізми тестування випробувального стенду

Через різноплановий характер IoT дуже складно розробити випробувальний стенд для загального аналізу безпеки. Можливості випробувального стенду повинні охоплювати різні стандарти зв'язку, такі як Wi-Fi, Bluetooth, Zigbee і т. п., а також вирішувати різні інші питання, пов'язані з іншими протоколами (з акцентом на протоколи зв'язку IoT і протоколи додатків). Крім того, системи, що використовуються пристроями IoT, варіюється, і їх функціональні можливості варіюються від простих до складних. Отже, мати можливість тестувати будь-який пристрій IoT, незалежно від його можливостей і характеристик, є складним завданням.

Крім того, випробувальний стенд повинен бути не просто орієнтований на один домен (наприклад, випробувальний стенд, призначений для бездротових сенсорних мереж); необхідно розробити багатодоменний тестовий пристрій для тестування різних типів пристроїв IoT.

Можливості тестування повинні обмежуватися не тільки тестуванням на проникнення, але також охопити різні інші методи аналізу (наприклад, інструменти для виявлення аномалій, інструмент для забезпечення стійкості до DoS і т. п.), що зробить випробувальний стенд для аналізу безпеки IoT ще більш інноваційним і здатним вирішувати інші завдання і складні питання. Ці інноваційні інструменти тестування безпеки демонструють проблеми і складність тестування безпеки пристроїв IoT.

Випробувальний стенд повинен складатися в основному з трьох механізмів, які використовуються для запуску і підтримки аналізу безпеки (Рисунок 3.4). Три механізму взаємодіють один з одним і використовуються для забезпечення функціональності тестового стенда. Пристрої IoT, вимірювальні

інструменти, точка доступу також є частиною комплексної установки випробувального стенду.

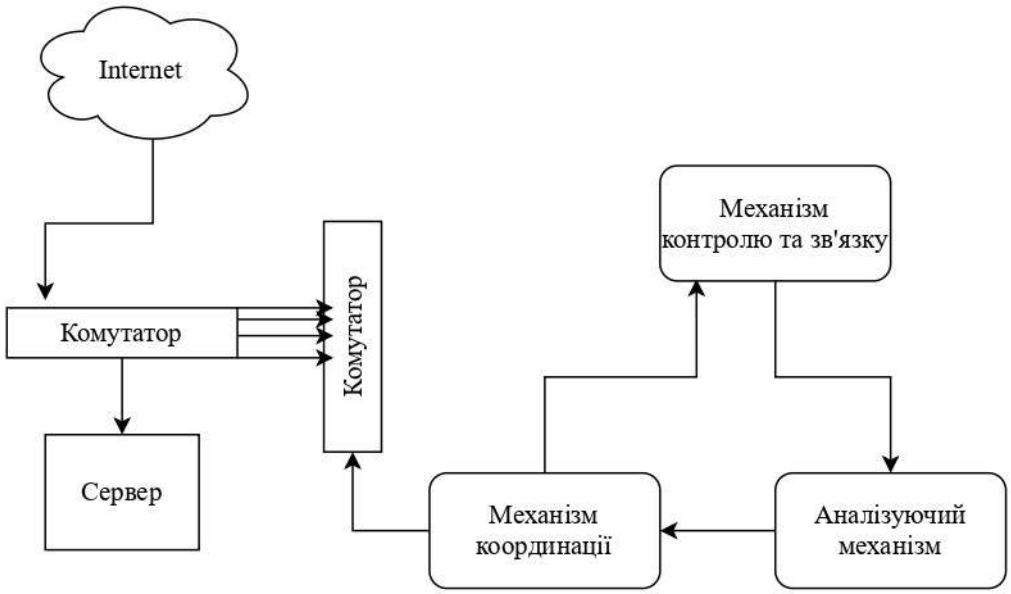


Рисунок 3.4 – Механізми тестування випробувального стенду

Три механізми полягають у наступному:

1. Механізм координації (СМ): розташований поза екранованої кімнати. СМ запускає пакет ПО TestStand, який виступає в якості організатора для запуску і генерації звіту після тесту;

2. Механізм контролю та зв’язку (ССМ): у ССМ працює LabVIEW NI, і пристрої IoT підключаються до ССМ для таких цілей, як включення/вимикання пристрою IoT, управління живленням, вимір споживаної потужності і т. д.;

3. Аналізуючий механізм: мета АМ - запустити інструменти тестування, такі як Nmap, Wireshark і т. д., необхідні для підтримки різних тестових випадків;

Всі три механізми взаємопов'язані один з одним.

Інструменти і механізми тестування безпеки, використовувані на випробувальному стенді (як інструменти з відкритим вихідним кодом, такі як Metasploit, так і розроблені механізми), генерують звіти в різних форматах даних, таких як *.csv, *.txt і т. д. Формат звіту залежить від використовуваного

інструменту. СМ об'єднує всі звіти для створення єдиного звіту в одному форматі даних, який виконується власним аналізатором. Користувач повинен мати доступ до звіту в будь-який час, а також може отримати звіт в різних інших форматах даних.

3.5 Опис використовуваного програмного забезпечення

3.5.1 TestStand

Спеціалізоване програмне забезпечення TestStand (Рисунок 3.5) розроблено для швидкої автоматизації процедури проведення випробувань і призначене для інженерів-випробувачів, які працюють на мовах програмування Python, C ++, G (в LabVIEW) та інших. Високотехнологічне середовище TestStand оснащене комплексними інструментами конфігурації, за допомогою яких можливо змінити і персоналізувати користувальницький інтерфейс, зробити генерацію звітів і налаштувати послідовність виконання тестів. Завдяки використанню середовища TestStand на підприємстві, у інженерів з’являється можливість повністю сфокусуватися на процесі тестування продукції, в той час, як система TestStand візьме на себе виконання процесів випробувань в певній послідовності, створюючи звіти про виконані роботи.

Переваги:

- Створення послідовності тестування;
- Створення інсталяторів для спрощення розгортання системи;
- Паралельне тестування і мінімізація витрат на обладнанні;
- Генерація звітів в різних форматах;
- Високоефективна технічна підтримка.

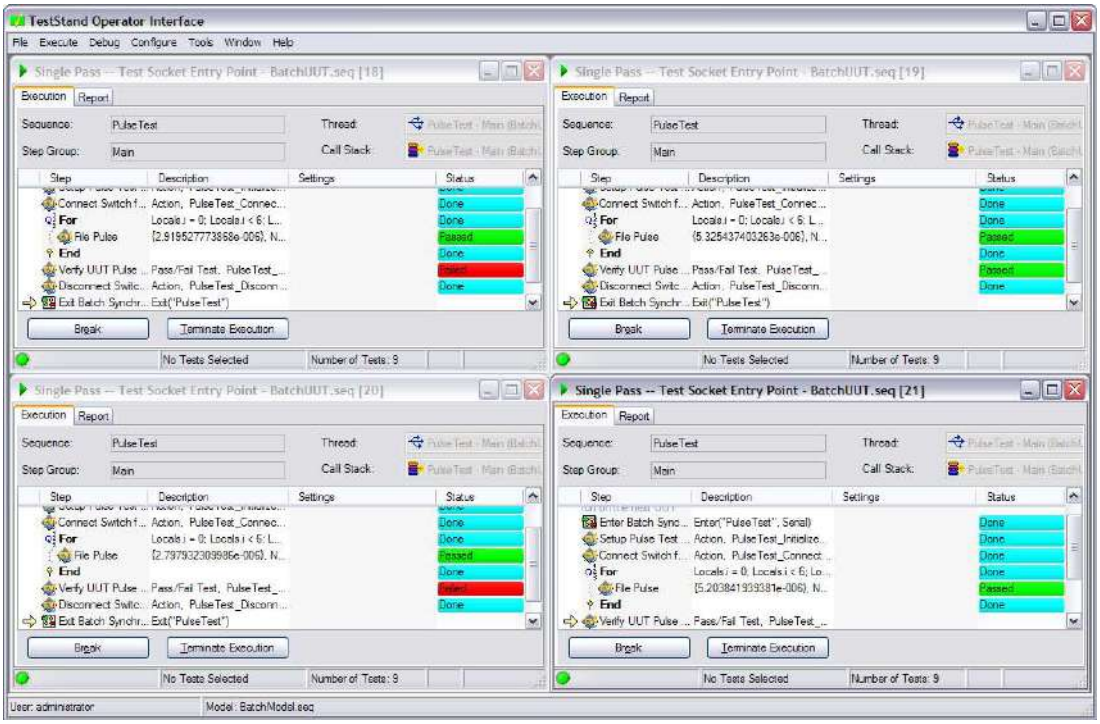


Рисунок 3.5 – Інтерфейс програмного середовища TestStand

4.5.2 LabView

LabVIEW - це спеціалізоване, гнучке середовище програмування, яке використовується в якості створення унікальних утиліт і додатків для приладів вимірювання (Рисунок 3.6). LabVIEW представляє із себе систему аналізу даних, яка включає в себе потужні інструменти для візуалізації результатів. За допомогою використання середовища LabVIEW інженери і вчені зможуть істотно збільшити продуктивність, скоротити час тестування, а також втілити в життя безліч різних бізнес-ідей при розробці додатків завдяки використанню графічного програмованого синтаксису.

Переваги:

- Швидкий старт з LabView;
- Гнучкість і масштабованість;
- Сумісність з іншими виробниками;
- Економія часу та коштів;
- Підвищення продуктивності тестових систем;
- Підтримка новітніх технологій для персонального комп'ютера.

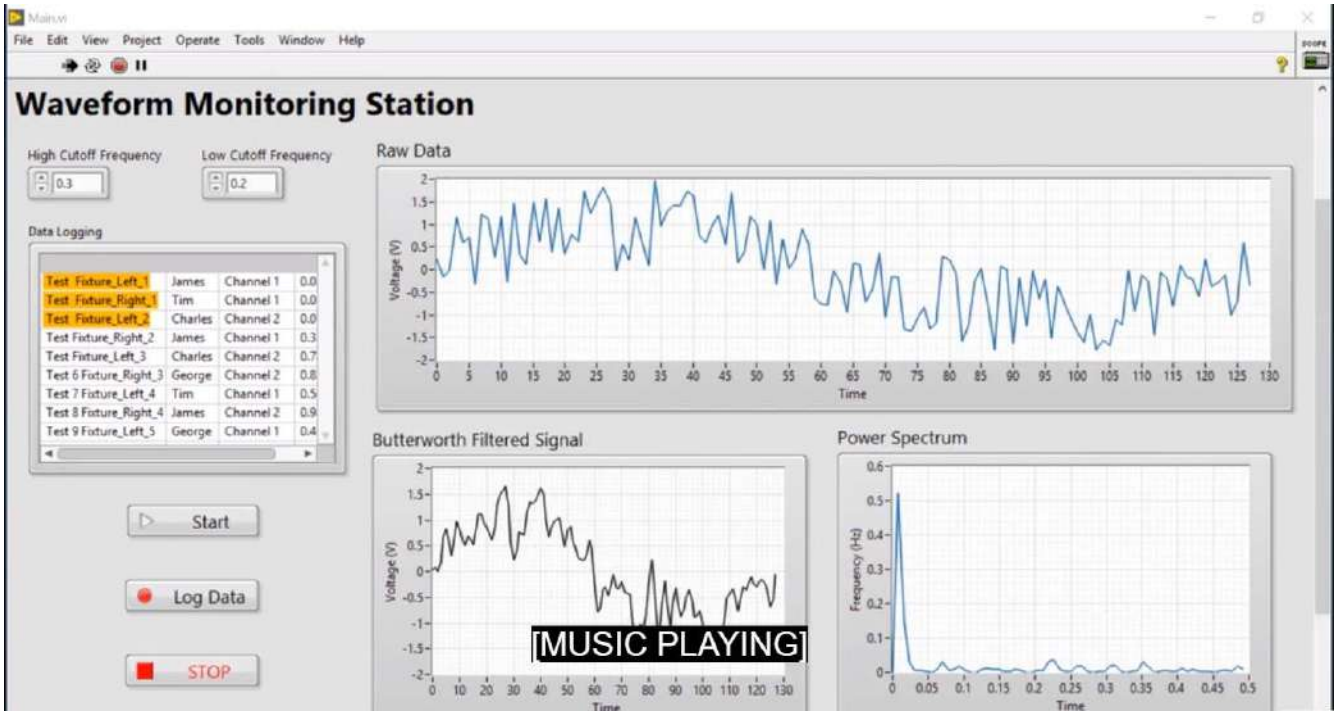


Рисунок 3.6 – Інтерфейс середовища програмування LabView

3.5.3 Nmap

Nmap - вільна утиліта, призначена для різноманітного настроюваного сканування IP-мереж з будь-якою кількістю об'єктів, визначення стану об'єктів сканованої мережі (портів і відповідних їм служб) (Рисунок 3.7). Спочатку програма була реалізована для систем UNIX, але зараз доступні версії для безлічі операційних систем.

Опції для проведення сканування:

- Визначення мети сканування;
- Виявлення хостів;
- Визначення портів і порядку сканування;
- Визначення служб і їх версій;
- Сканування з використанням скриптів;
- Визначення ОС;

- Обхід міжмережевих екранів, систем IDS;
- Виведення результатів.

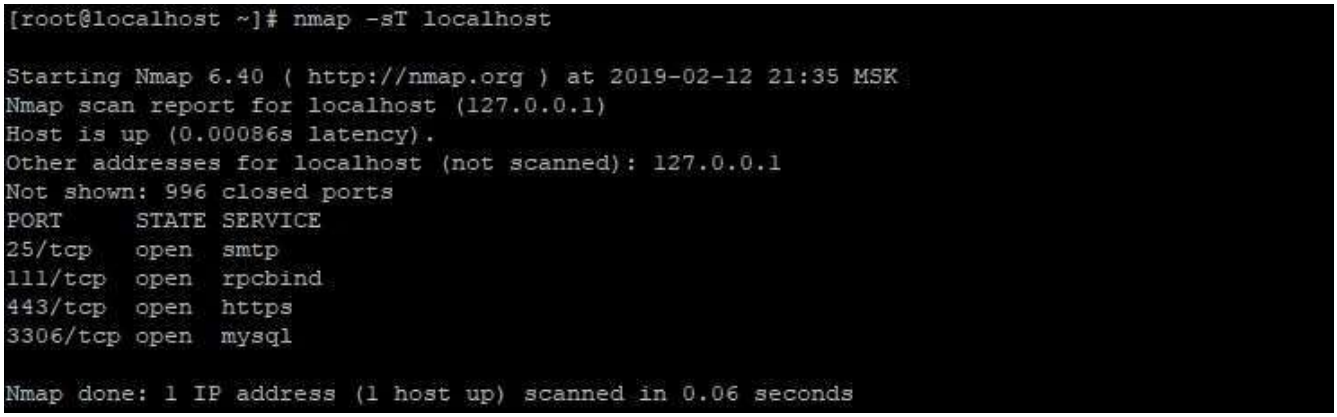


Рисунок 3.7 – Приклад сканування портів за допомогою утиліти Nmap

3.5.4 Dhspdump

Dhcpdump - утиліта для діагностики та налагодження роботи DHCP. При наявності в мережі DHCP-сервера, іноді виникає необхідність детального аналізу вмісту DHCP запитів / відповідей. В цьому випадку відмінним помічником виступає утиліта dhcpdump. Вказавши в якості параметра мережевий інтерфейс, можна отримати розшифровку всіх зафіксованих на ньому DHCP-пакетів.

3.6 Тестова операція

У загальному випадку СМ (при роботі TestStand і MRM) запускає тест. Більш конкретно, є послідовність кроків, написаних в TestStand, яка ініціює тест, запитуючи ССМ (з запущеними LabVIEW і STMM) виконати інтенсивне сканування, щоб знайти пристрої IoT, присутні в екранованої кімнаті. По завершенні сканування результати відправляються з СММ в СМ. Результати будуть складатися зі списку пристроїв IoT і їх IP-адрес і MAC-адрес.

Користувач може вибрати будь-який пристрій IoT зі списку для подальшого тестування.

Як тільки пристрій IoT вибрано, наступний крок в послідовності повинен вибрати тест, який буде виконаний. CM відображає список доступних тестів, наприклад, дактилоскопія, сканування вразливостей і т. д., і користувач може вибрати один або кілька тестів для виконання з обраним пристроєм IoT. Як тільки пристрій IoT і тест(и) були визначені, CM відправляє інформацію в CCM, і CCM відправляє інформацію в AM з усією відповідною інформацією (включаючи IP-адреса), необхідної для виконання тесту. AM (який запускає інструменти тестування, STM і MAM) виконує тест, і по завершенню тесту AM збереже звіт на локальному сервері і повідомить CMM, що тест завершений. CM отримує звіт з CCM через FTP і дає користувачеві можливість завершити тест або переглянути детальний звіт. Докладний звіт відображається на CM. Оскільки звіт присутній на локальному сервері, користувач може отримати доступ до звіту в будь-який час.

3.7 Один з можливих сценаріїв тестування

З використанням певного програмного забезпечення можна спробувати описати послідовність кроків для сканування портів. Слід ще раз коротко та точно вказати умови для можливості перевірки:

1. Створення графічного користувальницького інтерфейсу (GUI) у середовищі LabView (мова програмування – Python).
2. Створення програмних модулів для різних сценаріїв тестування безпеки IoT.
3. Підключення програмних код-модулів до середовища TestStand.
4. Включення IoT-пристрою в мережу.
5. Вибір необхідного тесту безпеки.

6. В залежності від сценарію буде відбуватися тестування за допомогою спеціальних програмних інструментів та утиліт.

7. Збереження звіту у форматі .XML або .HTML у базі даних (який можна відкрити у текстовому редакторі).

Метою сканування портів є дослідження виявлення пристроїв IoT шляхом спостереження за бездротовими/дротовими каналами зв'язку. Зокрема, сканування портів намагається визначити наявність пристрою і виявити відкриті і вразливі порти. Звіт про сканування портів також надає рівень ризику для кожного виявленого порту.

Послідовність кроків:

1. За допомогою графічного інтерфейсу аналізуючий механізм (при використанні TestStand) запускає утиліту Nmap для виявлення відкритих портів через настройку SSH на вибраному пристрої IoT.

2. Після того, як Nmap завершить сканування порту, результати зберігаються у вигляді файлу XML.

3. Користувальницький скрипт Python на АМ буде використовуватися для отримання списку відкритих портів, виявлених з файлу XML.

4. XML-файл зациклюється построково, перевіряючи ключове слово «Discovered / Open»

5. Будь-який рядок, який містить ключове слово «Discovered / Open», додається в файл, який буде містити список відкритих портів.

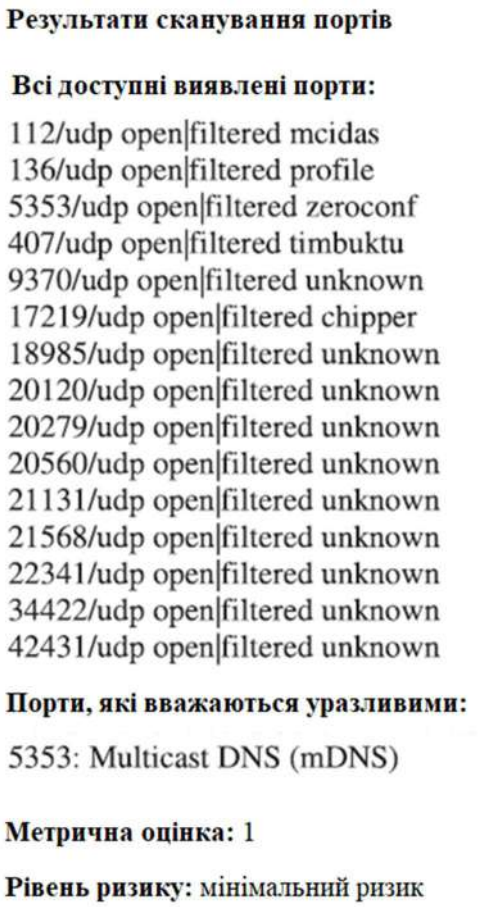
6. Нарешті, користувальницький скрипт Python порівнює відкриті порти зі списком найбільш вразливих відкритих портів і ідентифікує вразливі порти для створення звітів. Якщо слово «Discovered» не знайдено в XML-файлі, весь XML-файл копіюється в якості вихідного результату, який відображає все сканування.

Рекомендації до цього сценарію: для отримання списку найбільш вразливих відкритих портів рекомендується сервіс Tenable.io Security Center. Також варто прописати у скрипті метричну оцінку безпеки відкритих портів. Наприклад, рівень ризику може встановлюватися наступним чином: 0 -

безпечний, менше 15 - мінімальний ризик, від 15 до 30 - основний ризик та більше 30 - критичний ризик.

Загальні результати містять список відкритих портів, порти, які вважаються вразливими, і оцінки метрик.

На рисунку 4.1 показано, як може виглядати звіт з оцінки відкритих уразливих портів:


Результати сканування портів

Всі доступні виявлені порти:

112/udp open|filtered mcidas
136/udp open|filtered profile
5353/udp open|filtered zeroconf
407/udp open|filtered timbuktu
9370/udp open|filtered unknown
17219/udp open|filtered chipper
18985/udp open|filtered unknown
20120/udp open|filtered unknown
20279/udp open|filtered unknown
20560/udp open|filtered unknown
21131/udp open|filtered unknown
21568/udp open|filtered unknown
22341/udp open|filtered unknown
34422/udp open|filtered unknown
42431/udp open|filtered unknown

Порти, які вважаються уразливими:

5353: Multicast DNS (mDNS)

Метрична оцінка: 1

Рівень ризику: мінімальний ризик

Рисунок 4.1 – Вигляд звіту відображення уразливих портів

3.8 Висновки за Розділом 3

У цьому розділі розробляється платформа для тестування безпеки, спеціально призначена для пристроїв IoT. Пропонований випробувальний стенд для захисту має справу з найбільшою проблемою в області досліджень IoT, а саме з операціями тестування і аналізу безпеки, оскільки системи IoT

вважаються дуже складними середовищами через діапазон функціональних можливостей і різноманітность операцій, залучених до процесу. Аналіз ризиків безпеки і конфіденційності пристроїв IoT і їх впливу на існуючі системи / середовища вважається надзвичайно складним завданням через їх різнорідну природу, кількість типів пристроїв, а також безліч різних постачальників, технології і використовувані операційні системи, і можливості підключення, і той факт, що ці інтелектуальні пристрої використовуються в багатьох контекстах і станах. Випробувальний стенд може бути розширено новими тестами безпеки, симуляторами і стимуляторами, а також механізмами моніторингу та розширеного аналізу, які використовуються в якості внутрішніх та зовнішніх модулів випробувального стенду.

Пропонований випробувальний стенд вирішує завдання тестування різних пристроїв і конфігурацій IoT, включаючи пропрієтарні операційні системи і додатки з відкритим вихідним кодом, тому необхідні спільні і легко оновлювані механізми тестування безпеки, використовувані в випробувальному стенді. Крім того, оскільки пристрої IoT розробляються у вигляді закритих систем (як апаратних, так і програмних), випробувальний стенд для забезпечення безпеки націлений на вирішення проблеми тестування будь-якого вбудованого пристрою з використанням комбінації стандартних механізмів тестування безпеки поряд з передовими інструментами моніторингу та аналізу, щоб перевірити як внутрішній стан IoT-пристроїв (включаючи завантаження ЦП, споживання пам'яті, роботу файлової системи і т. д.), так і вплив пристрої на середу, в якій розгорнуто пристрій IoT (в основному з використанням операції аналізу трафіку).

4 ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ТА ОХОРОНИ ПРАЦІ

4.1 Аналіз умов праці в роботі з персональним комп'ютером

Робоче місце співробітника знаходиться в приміщенні, загальна площа якого становить 20 м2. Приміщення знаходиться на третьому поверсі будівлі. До основних шкідливих і небезпечних факторів виробничого середовища, які пов'язані безпосередньо з роботою на персональному комп'ютері та інших видах оргтехніки відносяться: випромінювання різного виду (рентгенівське, ультрафіолетове, інфрачервоне, електромагнітне і електростатичні поля), механічні шуми, пов'язані з роботою принтерів і вентиляторів, іонізація повітря, виділення в повітря робочого приміщення різних хімічних речовин (озон, трифенілфосфата, біфеніли), напруга зорових органів і пов'язане з ним перевтоми, значне навантаження на кисті рук і пальці, тривале перебування в одній і тій же позі, що викликає застійне явище в організмі , виділення з кондиціонерів різних мікроорганізмів. Крім того, мобільний телефон є джерелом електромагнітного випромінювання, на яке організм реагує певним чином.

Трудовий процес, пов'язаний з роботою за комп'ютером, істотно впливає на психофізіологічні можливості. Це пояснюється наступним:

- значне статичне фізичне навантаження;
- недостатня рухова активність;
- напруга сенсорного апарату, вищих нервових центрів, що впливають на функції уваги, мислення, регуляцію рухів.

Крім того, трудовий процес характеризується значними інформаційними навантаженнями.

Всі вищевказані фактори негативно впливають на здоров'я і сприяють виникненню професійних захворювань:

- комп’ютерний зоровий синдром, при якому проявляються такі симптоми: затуманення і зниження зору, порушення акомодації, швидке стомлення при читанні, печіння в очах, сильні болі при русі очей;
- виразки шлунка і нервовий стан;
- комп’ютерна алергія, яка проявляється у вигляді шкірних захворювань від трифінілфосфата;

- радиоволнова хвороба. При цій хворобі погіршується пам’ять, з’являються сильна дратівливість, серцево-судинні та нервові захворювання;

- кистьовий тунельний синдром;

- електромагнітне випромінювання впливає на нервову систему, проявляється в ослабленні пам’яті, сприйняття інформації, безпричинної втоми, безсоння, депресії і головних болях.

При регулярному використанні мобільного телефону спостерігаються такі наслідки:

- головні болі;
- зниження уваги;
- напруга в барабанних перегородках;
- ослаблення пам'яті;
- порушення сну та ін.

Іншим шкідливим фактором на робочому місці користувача є статична електрика. Накопичену статичну електрику, зокрема на екрані монітора, притягує пил, бруд та інші частинки, що містяться в повітрі. При цьому електризується не тільки екран, але і повітря на робочому місці, а також одяг з синтетичного або вовняного матеріалу.

4.2 Заходи щодо поліпшення умов праці

4.2.1 Поліпшення умов праці при роботі за комп'ютером

Для зменшення шкідливого впливу негативних факторів виробничого середовища необхідно вжити заходів щодо поліпшення умов праці за комп'ютером. Перед початком роботи необхідно пройти обстеження у лікаря-офтальмолога на предмет можливості здійснення роботи за комп'ютером, а також щорічно перевіряти стан органів зору. Під час роботи на персональному

комп'ютері, працівники повинні дотримуватися режиму праці та відпочинку, кілька разів на день робити своєрідну гімнастику для очей, наприклад таку:

- 10-20 разів заплющити очі і покліпати очима;
- «намалювати» вісімку (вертикальну і горизонтальну) закритими очима;
- помасажувати скроні (правий кут очей і лівий);
- для розслаблення м'язів кришталика ока зафіксувати погляд на найвіддаленішій точці (не менше 3 м), потім повільно перевести погляд на кінчик носа (10-15 разів).

Не допускається розташування робочого місця для роботи на комп'ютерах в підвальних і цокольних приміщеннях. Кімната повинна мати природне бічне освітлення, на вікнах повинні бути жалюзі або штори. Робочі столи слід розміщувати таким чином, щоб монітори були орієнтовані бічною стороною до світлових прорізів, щоб природне світло падало переважно ліворуч.

Приміщення, де розміщуються робочі місця повинні бути обладнані захисним заземленням. Заборонено розміщувати робочі місця з комп'ютером поблизу силових кабелів, високовольтних трансформаторів, технологічного обладнання, створює перешкоди в роботі комп'ютера.

Користувачі комп'ютерів повинні дотримуватися режиму праці та відпочинку (захист часом): обов'язково через кожні 2:00 години роботи робити 15-хвилинну перерву. Оптимальна відстань від очей користувача до екрану монітора приведена в таблиці 4.1. Площа, виділена для одного робочого місця з ПК повинна бути не менше 6 м^2, а об'єм - не менше 20 м^3.

Таблиця 4.1 - Рекомендації по розміщенню користувача щодо екрану

Розмір екрану по діагоналі		Відстань від очей користувача до екрану, мм
В дюймах (")	В сантиметрах	
14"/15"	35-38	500-600
17"	43	600-700
19"	48	700-800

21"	53	800-900

Заходи попередження впливу на зір користувачів ПК несприятливих умов на робочому місці:

- рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300-500 лк;
- на рівні екрану монітора доцільно встановити рухомий пюпітр (тримач документів), щоб зменшити "розкидання" точки зору;
- монітор ПК повинен відповідати вимогам шведських стандартів TCO'03;
- оптимальна відстань від очей користувача до екрану монітора повинна відповідати даним наведеним в таблиці;
- монітор повинен бути розташований щодо джерел світла (вікон, світильників) таким чином, щоб світло падало збоку, переважно зліва, і на екрані не було відблисків від них;
- для штучного місцевого освітлення бажані світильники зі спеціальними матовими лампочками;
- якщо є можливість, то необхідно використовувати монітор з плоским екраном.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання. Висота робочої поверхні столу повинна бути 725 мм, робоча поверхня стола повинна мати ширину 800 .. 1400 мм і глибину 800 .. 1000 мм. Робочий стіл повинен мати простір для ніг висотою не менше 600 мм, шириною - не менше 500 мм, глибиною на рівні колін - не менше 450 мм і на рівні витягнутих ніг - не менше 650 мм. Поверхня робочого столу повинна мати коефіцієнт відбиття 0,5 - 0,7.

Клавіатуру слід розташовувати на поверхні столу на відстані 100 - 300 мм від краю, оберненого до користувача, або на спеціальній, регульованою по висоті робочої поверхні, яка відокремлена від основної стільниці. Конструкція робочого стільця (крісла) повинна забезпечувати підтримку раціональної

робочої пози під час роботи за комп'ютером, дозволяти змінювати позу з метою зниження статичного напруження м'язів шийно-плечової області і спини для попередження розвитку втоми.

Розташування робочого місця з ПК на малюнку 4.1.

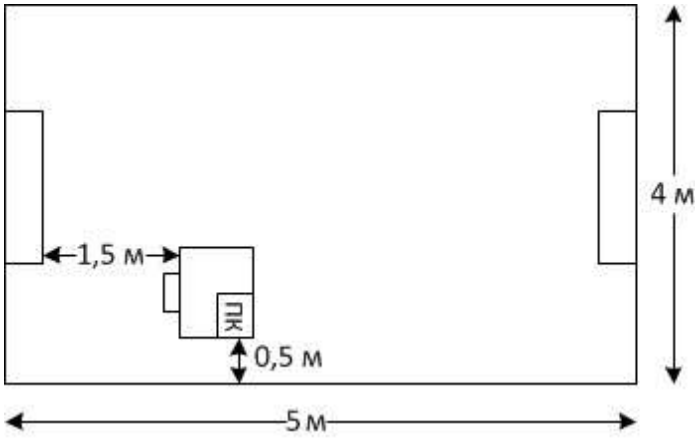


Рисунок 4.1 - Облаштування робочого місця з ПК

При аналізі облаштування робочого місця з ПК виявлені наступні порушення: робочий стіл, обладнаний з ПК знаходиться навпроти вікон, таким чином працюючий звернений до вікна, що заборонено; робоче місце з ПК розташоване від стіни на відстані 0,5 м, а воно має розташовуватися на відстані не менше 1 м.

Відповідно до проведених досліджень рекомендується провести наступні заходи щодо поліпшення робочого місця: перемістити робоче місце від стіни на відстань 1м; поставити стіл таким чином, щоб світло падало на нього збоку. Таким чином робоче місце користувача ПК буде розташований згідно ГОСТ і матиме вигляд представлений на малюнку 4.2.

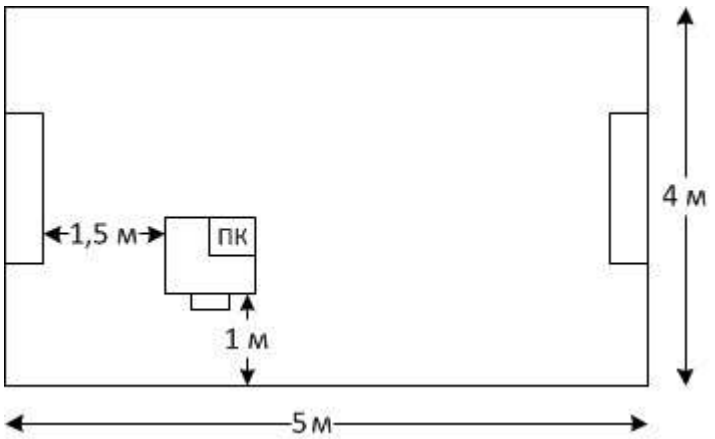


Рисунок 4.2 - Облаштування робочого місця з ПК відповідно до ГОСТ

4.2.2 Мікроклімат робочого місця

Мікроклімат (температура, відносна вологість і швидкість руху повітря) і рівень іонізації повітря на робочих місцях користувачів повинні відповідати вимогам нормативних документів, зокрема, в приміщеннях з ПК повинен бути забезпечений трикратний обмін повітря в годину, адже під час розумової праці мозок людини споживає в 9 -10 разів більше кисню, ніж в звичайних умовах.

Рекомендується застосовувати припливно-витяжну вентиляцію або систему кондиціонування повітря, прилади зволоження та / або генерації негативних іонів (аероіонізатори). Краще, щоб всі ці операції (очищення повітря, аероионизация, зволоження, підтримання необхідної температури) забезпечувалися кондиціонером.

У робочому приміщенні з комп'ютерами є джерела тепловиділення, тому необхідно визначити умови вентилювання.

Обмін повітря по теплу визначається за формулою:

$$L = \frac{Q_{збд}}{c * p(t_E - t_H)}, \frac{м^2}{ч} \qquad (4.1)$$

де Qзбд - надлишкове виділення тепла в робочому приміщенні, ккал / год .;

с - теплоємність повітря (0,237 ккал / кг);

р - об'ємна маса повітря (1,226 кг / м3);

$t_в$ - температура витяжного повітря (30 ° С);

$t_н$ - температура припливного повітря (20 ° С).

Розрахуємо надлишкове виділення тепла за формулою:

$$Q_{изб} = Q_{об} + Q_{пер} + Q_{осв} + Q_{ср} \quad (4.2)$$

де $Q_{об}$ - виділення тепла від устаткування;

$Q_{пер}$ - виділення тепла співробітниками;

$Q_{осв}$ - виділення тепла від електричного освітлення;

$Q_{ср}$ - виділення тепла від сонячної радіації через вікна.

Визначимо виділення тепла від устаткування за формулою:

$$Q_{уст} = P * K_а * K_б * 860, \quad (4.3)$$

де Р - сумарна потужність обладнання, кВт / год;

$К_а$ - коефіцієнт питомої потужності (0,95);

$К_Б$ - коефіцієнт одночасної роботи (1,0).

Р визначається за такою формулою:

$$P = (\chi_1 * k_1) + (\chi_2 * k_2) + (\chi_3 * k_{31}) + (\chi_4 * k_4), \quad (4.4)$$

де x_1, x_2, x_3, x_4 - кількість системних блоків, моніторів, принтерів, кондиціонерів відповідно;

k_1, k_2, k_3, k_4 - їх потужність.

Розрахуємо:

$$Q_{уст} = \left[(1 * 0,5) + (1 * 0,1) + (1 * 0,4) + (1 * 6,3)\right] * 0,95 * 1 * 860 = 5964,1 ккал / ч$$

Визначимо виділення тепла від персоналу за допомогою формули:

$$Q_{пер} = n * g, \quad (4.5)$$

де n - кількість співробітників;

g - кількість тепла, виділяє один співробітник на годину (100 ккал / год)

Розрахуємо:

$$Q_{пер} = 1 * 100 = 100 ккал / ч$$

Визначимо надходження тепла від електричного освітлення за формулою:

$$Q_{ост} = E_т * g_1 * S, \quad (4.6)$$

де $E_м$ - нормована освітленість для цієї зорової роботи, приймаємо рівною 400 лк;

g1 - питоме тепловиділення на 1 м^2 підлоги при 1 лк освітленості (для люмінесцентних ламп - 0,05 ккал / год.).

S - площа приміщення, м^2.

Розрахуємо:

$$Q_{осв} = 400 * 0,05 * 20 = 400 ккал / ч$$

Визначимо надходження тепла від сонячної радіації через вікна за формулою:

$$Q_{ср} = F * g_2 * K_{осл}, \quad (4.7)$$

де F - площа віконних прорізів (3 м2).

g_2 - кількість тепла, що надходить через 1 м2 віконного отвору (65 ккал / ч).

$К_{осл}$ - коефіцієнт ослаблення, приймаємо рівним 0,4.

Розрахуємо:

$$Q_{ср} = 3 * 65 * 0,4 = 78 ккал / ч$$

Визначимо кількість надлишкового тепла:

$$Q_{над} = 6542,1 ккал / ч$$

Визначимо витрати повітря в приміщенні:

$$L = \frac{6542,1}{0,237 * 1,226 * (30 - 20)} = 2251,533 м^3 / ч$$

Отже, необхідна система кондиціонування і вентиляції з продуктивністю не менше 2500 м3 / год.

Параметри мікроклімату на робочих місцях регламентуються ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень». Згідно до даних санітарним нормам температура повітря, швидкість руху повітря і відносна вологість в холодні періоди року повинна становити 21-23 градуси за Цельсієм, 0,1 метра в секунду і 40-60% відповідно. У теплі періоди року температура повітря повинна складати 22-24 градусів Цельсія, рухливість повітря 0,2 метра в секунду, вологість 40-60%.

4.3 Пожежна безпека

Для споруд і приміщень, в яких експлуатуються комп'ютери, існують спеціальні протипожежні заходи, визначені Правилами пожежної безпеки. Однією з умов роботи є дотримання правил пожежної безпеки.

Будинки й ті їх частини, в яких розташовуються комп'ютери, повинні бути нижче II ступеня вогнестійкості. Над і під приміщеннями, де розташовуються комп'ютери, а також суміжних з ними приміщеннях дозволяється розташування приміщень категорій А і Б за вибухопожежною небезпекою. Приміщення категорії В слід відокремлювати від приміщень з комп'ютерами протипожежними стінами. Категорії приміщення повинні бути нанесені на вхідних дверях.

У приміщенні, де розташовано робоче місце співробітника, основні заходи для забезпечення пожежної безпеки визначає Інструкція про заходи пожежної безпеки для службових приміщень. Вона є обов'язковою для виконання всіма співробітниками.

В інструкції про засоби пожежної безпеки для службових приміщень забороняється:

- влаштовувати тимчасові електромережі, застосовувати саморобні плавкі вставки в запобіжниках, прокладати електричні дроти безпосередньо на горючій основі, експлуатувати світильники зі знятими ковпаками (розсіювачами), використовувати саморобні подовжувачі, які не відповідають вимогам Правил улаштування електроустановок;

- пристосовувати вимикачі, штепсельні розетки для підвішування одягу та інших предметів, обгортати електролампи і світильники, заклеювати ділянки електромережі горючою тканиною, папером;

- використовувати побутові електрокип'ятильники, чайники і т.п. без негорючих підставок, залишати без нагляду включеними в електромережу кондиціонери, комп'ютери, рахункові і друкарські машинки тощо.;

- захаращувати підступи до засобів пожежогасіння, використовувати пожежні крани, рукави і пожежний інвентар не за призначенням, зберігати документи, різні матеріали, предмети та інвентар в шафах (нішах) інженерних комунікацій;

- палити (крім спеціально відведених для цього адміністрацією місць, позначених написом «Місце для куріння» та забезпечених урною або попільничкою з негорючого матеріалу), проводити зварювальні та інші вогневі роботи без оформлення відповідного дозволу, застосовувати легкозаймисті рідини.

Приміщення, в яких розташовуються персональні комп'ютери, повинні бути оснащені системою автоматичної пожежної сигналізації з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 $м^2$ площі приміщення з урахуванням гранично допустимих концентрацій вогнегасної речовини. Не рідше одного разу на квартал необхідно очищати від пилу агрегати та вузли, кабельні канали та простір між статями. Електромережі, електроприлади і апаратура повинні експлуатуватися тільки в справному стані. У разі виявлення пошкоджень електромереж, вимикачів, розеток ті інших електровиробів слід негайно

відключити їх і вжити необхідних заходів до приведення їх у пожежобезпечний стан.

4.4 Безпека при надзвичайних ситуаціях на підприємстві

Об'єктом розгляду на предмет визначення надзвичайних небезпек і їх наслідків є будівля, в якому розташований працівник з комп'ютером. Однією з ймовірних загроз може бути раптове виникнення пожежі в результаті короткого замикання в електромережах або розрядів статистичного електрики, що може привести до пошкодження і руйнування будівлі, обладнання, комунікацій, виділення токсичних продуктів горіння. Тому для цієї будівлі повинен бути розроблений оперативний план гасіння пожежі, який визначає порядок дії персоналу при пожежі, порядок його гасіння в електроустановках, взаємодія з власним складом пожежних підрозділів, а також застосування схем і засобів пожежогасіння з урахуванням заходів безпеки.

Будівля має бути обладнано мережею протипожежного водопостачання, установками виявлення і гасіння пожеж відповідно до вимог нормативно-технічних документів. Кожен працівник повинен чітко знати і виконувати вимоги правил пожежної безпеки та протиаварійного режиму на об'єкті, вміти користуватися наявними вогнегасниками, іншими первинними засобами пожежогасіння та знати місце їх розташування.

Меблі та обладнання повинні розміщуватися таким чином, щоб забезпечувався вільний евакуаційний прохід до дверей виходу з приміщення (завширшки не менше 1 м). Евакуаційні шляхи і виходи необхідно постійно тримати вільними, нічим не захаращувати. Засоби протипожежного захисту в приміщеннях повинні триматися в справному стані. У разі виявлення пожежі слід: негайно повідомити державну пожежну охорону за телефоном «101», вказати при цьому адресу, кількість поверхів, місце виникнення пожежі, наявність людей, своє прізвище; повідомити про пожежу керівництву, а в

нічний час черговому охоронцю; по можливості почати гасіння пожежі наявними засобами, організувати зустріч пожежних підрозділів.

При виникненні пожежі в початковій стадії його розвитку випромінюється тепло, накопичуються токсичні продукти згоряння, можливі обвалення будівельних споруд. Тому слід якомога швидше провести евакуацію людей з палаючої будівлі. Показником ефективності евакуації є час, протягом якого працівники можуть при необхідності залишити окремі приміщення і будівля в цілому. Безпека евакуації досягається тоді, коли час евакуації не перевищує час настання критичної фази розвитку пожежі, тобто часу від початку пожежі до досягнення граничних для людини впливів факторів пожежі (критичних температур, ступеня задимлення, зниження концентрації кисню і т.п.). Число евакуаційних виходів повинно бути не менше двох. Вони повинні розташовуватися розосереджено.

Двері на шляхах евакуації повинні відкриватися у напрямку виходу з будівлі. У кожному приміщенні на видному місці повинен бути вивішений план евакуації при пожежі. Рухатися в задимленій зоні необхідно повзучи або максимально пригнувшись, так як більшість нагрітих газоподібних отруйних речовин і дим збираються у верхній зоні приміщення, крім цього, в приміщенні при горінні температура на рівні очей людини в 6 разів вища за температуру на рівні підлоги, до того ж внизу завжди зберігається велика концентрація кисню. Коли ви опинилися біля вікна трохи відкрийте його і дихайте через щілину, очікуючи прибуття пожежних. При їх прибуття негайно зверніть на себе увагу. Ніколи не стрибайте у вікно без крайньої на це необхідності (кожен другий стрибок з 4-го поверху під час пожежі смертельний).

4.5 Висновки за Розділом 4

В даному розділі зроблено аналіз умов праці співробітника, який працює за комп'ютером. Проведено заходи щодо поліпшення умов праці, виконаний розрахунок системи вентиляції, розглянуті заходи пожежної безпеки та безпеки при надзвичайних ситуаціях, оптимізовано розміщення робочого місця співробітника в приміщенні, відповідно до встановлених норм і ГОСТами.

ВИСНОВКИ

IoT - це нова технологія, яка перетворює звичайні фізичні пристрої, такі як телевізори, холодильники, годинники, автомобілі і т.д., в інтелектуальні підключені пристрої. Потенційні додатки, пов’язані з IoT, здаються нескінченними, а нові та інноваційні функції і можливості розробляються практично щодня. Однак великі переваги і можливості, що надаються комп’ютерами IoT, супроводжуються серйозними потенційними ризиками для безпеки і конфіденційності. Більш того, через різнорідну природу таких пристроїв (різних типів пристроїв з різними конфігураціями програмного і апаратного забезпечення, вироблених різними виробниками і т.д.), і того факту, що вони використовуються в різних контекстах, аналіз і забезпечення безпеки таких пристроїв вважається складним завданням.

У цій роботі розглянуто детальну архітектуру IoT для кращого розуміння роботи екосистеми IoT, розглянуто основні проблеми безпеки на кожному рівні архітектури, також пропонуються контрзаходи для різних проблем безпеки для захисту систем IoT, обговорюються проблеми, пов'язані із застарілими рішеннями безпеки в IoT.

У даній роботі розробляється інструментарій тестування безпеки, призначений для IoT-пристроїв. Тестове середовище безпеки призначено для тестування всіх типів IoT-пристроїв, з різними програмними / апаратними конфігураціями, шляхом проведення тестування безпеки. Для моніторингу роботи тестованого IoT-пристрою в цілому в тестовій платформі використовуються вдосконалені процеси аналізу, засновані на алгоритмах машинного навчання. Обговорюється архітектурний проект пропонованого стенду.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гольдштейн Б.С., Кучерявый А.Е. Сети связи пост-NGN
2. https://www.hindawi.com/journals/jece/2017/9324035/
3. I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, and D. P. Agrawal, “Choices for interaction with things on Internet and underlying issues,” *Ad Hoc Networks*, vol. 28, pp. 68–90, 2015.
4. O. Said and M. Masud, “Towards internet of things: survey and future vision,” *International Journal of Computer Networks*, vol. 5, no. 1, pp. 1–17, 2013.
5. M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du, “Research on the architecture of internet of things,” in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE '10)*, vol. 5, pp. V5-484–V5-487, IEEE, Chengdu, China, August 2010.
6. R. Khan, S. U. Khan, R. Zaheer, and S. Khan, “Future internet: the internet of things architecture, possible applications and key challenges,” in *Proceedings of the 10th International Conference on Frontiers of Information Technology (FIT '12)*, pp. 257–260, December 2012.
7. M. Swan, “Sensor mania! The internet of things, wearable computing, objective metrics, and the quantified self 2.0,” *Journal of Sensor and Actuator Networks*, vol. 1, no. 3, pp. 217–253, 2012.
8. M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky, “Key ingredients in an IoT recipe: fog computing, cloud computing, and more fog computing,” in *Proceedings of the IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD '14)*, pp. 325–329, Athens, Greece, December 2014.
9. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *Proceedings of the 1st ACM MCC Workshop on Mobile Cloud Computing*, pp. 13–16, 2012.

10. D. Zeng, S. Guo, and Z. Cheng, “The web of things: a survey,” *Journal of Communications*, vol. 6, no. 6, pp. 424–438, 2011.

11. A. Francillon, C. Castelluccia, “Code Injection Attacks on Harvard-Architecture Devices.” ACM 2008

12. L. Li, "Study on security architecture in the Internet of Things." In Measurement, Information and Control (MIC), *2012 International Conference on, vol. 1, pp. 374-377*. IEEE, 2012.

13. A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs." *Communications Surveys & Tutorials, IEEE 11, no. 4 (2009): 42-56*.

14. B. Khoo, "RFID as an enabler of the internet of things: issues of security and privacy." In Internet of Things (iThings/CPSCom), *International Conference on and 4th International Conference on Cyber, Physical and Social Computing, pp. 709-712. IEEE, 2011*.

15. B. S. Thakur, and S. Chaudhary, "Content sniffing attack detection in client and server side: A survey." *International Journal of Advanced Computer Research (IJACR) 3, no. 2 (2013): 10.*

16. D. Wu, and G. Hu, "Research and improve on secure routing protocols in wireless sensor networks." In Circuits and Systems for Communications, 2008. ICCSC 2008. *4th IEEE International Conference on, pp. 853-856. IEEE, 2008.*

17. J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses." In Proceedings of the *3rd international symposium on Information processing in sensor networks, pp. 259-268. ACM, 2004.*

18. The treacherous 12, Cloud Computing top threats 2016, “Top threats working group, Cloud Security Alliance (CSA)”

19. T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer, "Social phishing." Communications of the ACM 50, no. 10 (2007): 94-100

20. H. Tobias, et al. "Security Challenges in the IP-based Internet of Things." *Wireless Personal Communications 61, no. 3 (2011): 527-542.*