КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій проєктування та прикладної математики

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Проєктування інформаційної системи кібербезпеки для інтернету речей»

ПЛОСКОНОС ІЛЛЯ ГЕННАДІЙОВИЧ

(прізвище, ім’я та по батькові студента повністю)

Київ 2026 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій проєктування та прикладної математики

(кафедра)

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТППМ
д.т.н., професор Бородавка Є.В.

_____
„___” _____2026 року

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: "Проєктування інформаційної системи кібербезпеки для інтернету речей"

Виконав: студент 4-го курсу, групи ІСТ-22

Спеціальності: 126 «Інформаційні системи та технології»

(шифр і назва спеціальності)

Плосконос І.Г.

(прізвище та ініціали)

Керівник доктор філософії Бугров А.А.

(прізвище та ініціали)

Рецензент доктор філософії Рябчун Ю.В.

(прізвище та ініціали)

Київ 2026 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

Факультет: автоматизації і інформаційних технологій
Кафедра: інформаційних технологій проєктування та ПМ
Освітній рівень: «бакалавр» за ОПП
Спеціальність: 126 «Інформаційні системи та технології»
Освітня програма: «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТППМ
д.т.н., професор Бородавка Є.В.

_____

„__” ___2026 року

З А В Д А Н Н Я
ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

Плосконос Ілля Геннадійович

1. Тема роботи: “Проєктування інформаційної системи кібербезпеки для інтернету речей”
затверджена наказом ректора КНУБА № 444/52-15/13.6/26 від “08” квітня 2026 року
2. Керівник роботи: Бугров Анатолй Анатолійович, доктор філософії, старший викладач кафедри ІТППМ
3. Строк подання студентом роботи до захисту: .червень 2026 року
4. Зміст пояснювальної записки за розділами:
   Р.1. Дослідження архітектури та технологій IoT.
   Р.2. Дослідження основних загроз безпеці в IoT.
   Р.3. Способи забезпечення кібербезреки в IoT.
5. Інформаційні слайди:
   С.1 Титульний слайд.
   С.2 Мета, завдання, актуальність роботи.
   С.3 Визначення інтернету речей.
   С.4 Архітектура IoT.
   С.5 Характеристика можливих атак на IoT.
   С.6 Криптографічні методи. Гібридне шифрування.

С.7 Системи виявлення та запобігання вторгненням.
С.8 Методологія проведення дослідження.
С.9 Тестування систем виявлення та запобігання вторгненням.
С.10 Висновки.

6. Календарний план виконання кваліфікаційної роботи

Види робіт та їх зміст	Дата виконання
Р. 1. Дослідження архітектури та технологій IoT	Лютий 2026 р.
Р. 2. Дослідження основних загроз безпеці в IoT	Лютий 2026 р.
Р. 3. Способи забезпечення кібербезреки в IoT	Березень 2026 р.
Висновок	Квітень 2026 р.
Остаточне оформлення роботи	Травень 2026 р.
Попередній захист роботи на кафедрі	Червень 2026 р.

7. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	Дата	Підпис
Прийом програмного продукту	Доктор філософії, доцент Рябчун Ю.В.		

8. Дата видачі завдання: 12 січня 2026 року

Керівник _____ Бугров А.А.
(підпис) (прізвище та ініціали)

Бакалавр _____ Плосконос І.Г.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

«Проєктування інформаційної системи кібербезпеки для інтернету речей».

Кваліфікаційна робота бакалавра за спеціальністю: 126 «Інформаційні системи та технології», освітня програма: «Інформаційні системи та технології». – Київський національний університет будівництва та архітектури. – Київ, 2026.

Актуальність даної теми обумовлена тим, що інтернет речей, на даний час – одна з найважливіших течій розвитку інтернету, та загалом всіх інформаційних технологій. Це збільшує вразливість систем інтернету речей з точки зору кібербезпеки. Для того щоб забезпечити кібербезпеку в цих системах, потрібно провести певні організаційні заходи та впровадити технічні засоби забезпечення кібербезпеки.

Ключові слова: Інтернет речей, кібератака, розумний пристрій, вторгнення.

SUMMARY

" Project of cybersecurity system when using the Internet of Things".

Bachelor's qualification work in the specialty: 126 "Information systems and technologies", educational program: "Information systems and technologies". - Kyiv National University of Construction and Architecture. - Kyiv, 2026.

The relevance of this topic is due to the fact that the Internet of Things is currently one of the most important trends in the development of the Internet, and in general of all information technologies. These factors increase the vulnerability of the Internet of Things for cybersecurity. In order to ensure cybersecurity in these systems, it is necessary to carry out certain organizational measures and implement technical facilities to ensure cybersecurity.

Keywords: Internet of Things, cyberattack, smart device, invasion.

ЗМІСТ

ВСТУП

У роботі розглянуто проблему захисту інформації при використанні технологій інтернету речей.

Інтернет речей знаходиться лише на початку свого шляху, але вже розвивається з величезною швидкістю, і всі впроваджені нововведення створюють нові проблеми, пов'язані з інформаційною безпекою.

Сьогодні пристрої Інтернету речей не лише масово використовуються у щоденному вжитку, але й у сучасному бізнес-середовищі. Зокрема Інтернет речей активно впроваджується в різних галузях — від промислової сфери до сільського господарства, рітейлу та будівництва. Поступово пристрої IoT стають невід’ємною частиною багатьох бізнес-процесів, і зростання їх кількості спричиняє виникнення нових проблем безпеки.

Ми можемо знайти пристрої IoT майже скрізь:

- В системах опалення, вентиляції, кондиціонування;
- В енергетичному секторі - вітряки, генератори, лічильники, турбіни, паливні елементи, бурові установки, тощо;
- В споживчому та побутовому секторі - побутова техніка, телевізори, ігрові приставки, сигналізація, замки, та інше;
- В промисловому секторі - конвеєри, монтажні установки, насоси, клапани, резервуари, гірничодобувна техніка, тощо;
- В транспорті - автомобілі, літаки, рекламні вивіски, ліхтарі, кораблі тощо;
- В секторі роздрібної торгівлі - етикетки, вивіски та торгові автомати;
- В секторі воєнної та громадської безпеки - машини швидкої допомоги, військові машини, датчики моніторингу навколишнього середовища;
- В IT секторі - комутатори, маршрутизатори, сховища даних, телефонні системи, сервери;
- В секторі охорони здоров’я - імплантати, хірургічне обладнання, монітори, насоси, МРТ;

- В сільськогосподарському секторі - метеостанції, пристрої управління посівами, ошийники для тварин, сільськогосподарські безпілотники, датчики в теплицях та вуликах, тощо.

Пристрої IoT зустрічаються в повсякденному житті набагато частіше, ніж ми уявляємо, і це робить вимоги до безпеки IoT, дуже важливим питанням для кожного з нас, а не лише для людей які користуються цією технологією.

Об’єктом роботи є екосистема інтернету речей, що включає розумні пристрої та мережі що їх поєднують

Метою роботи є дослідження будови та технологій інтернету речей, проведення аналізу протоколів і розробка способів забезпечення кібербезпеки в IoT.

1 ДОСЛІДЖЕННЯ АРХІТЕКТУРИ ТА ТЕХНОЛОГІЙ IOT

1.1 Основні поняття інтернету речей

Інтерне́т рече́й (англ. Internet of Things, IoT) — концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані давачі, а також програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами в автоматичному режимі, за допомогою використання стандартних протоколів зв'язку. Окрім давачів, мережа може мати виконавчі пристрої, вбудовані у фізичні об'єкти і пов'язані між собою через дротові чи бездротові мережі. Ці взаємопов'язані пристрої мають можливість зчитувати дані та приводити в дію механізми, а також зменшують необхідність участі людини, за рахунок використання інтелектуальних інтерфейсів.

Інтернет-речей грунтується на трьох базових принципах. По перше, на повсюди поширену комунікаційну інфраструктуру, по-друге, на глобальну ідентифікацію кожного об'єкта і, по-третє, на можливість кожного об'єкта відправляти і отримувати дані за допомогою персональної мережі або мережі Інтернет, до якої він підключений.

В IoT кожен пристрій має свій ідентифікатор, які спільно утворюють об’єднання пристроїв, здатних взаємодіяти один з одним, створюючи тимчасові або постійні мережі. Так пристрої можуть брати участь в процесі їх переміщення, ділячись інформацією про поточну геопозицію, що дозволяє повністю автоматизувати процес логістики, а маючи вбудований інтелект, пристрої можуть змінювати свої властивості і адаптуватися до навколишнього середовища.

Інтернет-пристрої мають єдиний протокол взаємодії, згідно якому будь-який вузол мережі рівноправний в наданні своїх сервісів. Кожен вузол мережі інтернет-речей надає свої послуги. У той же час вузол такої мережі може приймати команди від будь-якого іншого вузла. Це означає, що всі

інтернет-речі можуть взаємодіяти один з одним і вирішувати спільні обчислювальні завдання. Інтернет-речі можуть утворювати локальні мережі, об'єднані будь-якої однією зоною обслуговування.

1.2 Архітектура IoT

Як правило, системи IoT можна розділити на дві групи: ті, що застосовуються лише для моніторинга (як правило, вимагають великих обсягів даних), і ті, що мають на меті забезпечити дистанційне управління "інтелектуальними" пристроями.

Перші - це, як правило, аналітичні рішення, які збирають інформацію з безлічі датчиків IoT (дані надходять через шлюзи), а потім форматують або візуалізують їх, щоб допомогти користувачам розпізнати закономірності та мати уявлення що відбувається в системі. Вони можуть мати величезні вимоги до пропускної здатності, коли до них підключено багато пристроїв.

Мережі IoT другого типу, популярні серед підприємств, зазвичай складаються з безлічі об'єктів, і безпечне їх масштабування, впроваджуючи нові пристрої та потоки даних, є надзвичайно складним завданням. Вразливості накопичуються в геометричній прогресії, коли в мережу додаються нові елементи та відносини.

Не існує єдиного консенсусу щодо архітектури IoT, який визнаний всіми науковцями. Різні дослідники пропонували різні архітектури.

1.2.2 Тришарова архітектура

Базовою та найпоширенішою архітектурою є тришарова архітектура , яка зображена на рисунку 1. Вона була представлена на ранніх етапах досліджень у цій галузі. Вона складається з трьох шарів, а саме - рівень сприйняття, мережевий та прикладний рівні.

- *Рівень сприйняття* - це фізичний рівень, який має датчики для зондування та збору інформації про навколишнє середовище. Він розпізнає деякі фізичні параметри або визначає інші розумні об’єкти в навколишньому середовищі.

- *Мережевий рівень* відповідає за підключення до інших розумних девайсів, мережевих пристроїв та серверів. Його функції також використовуються для передачі та обробки даних датчиків.

- *Програмний рівень* відповідає за надання користувачеві конкретних послуг. Він визначає різні програми, в яких може бути розгорнутий Інтернет речей, наприклад, розумні будинки, розумні міста, та ін.

1.2.2 П’ятишарова архітектура

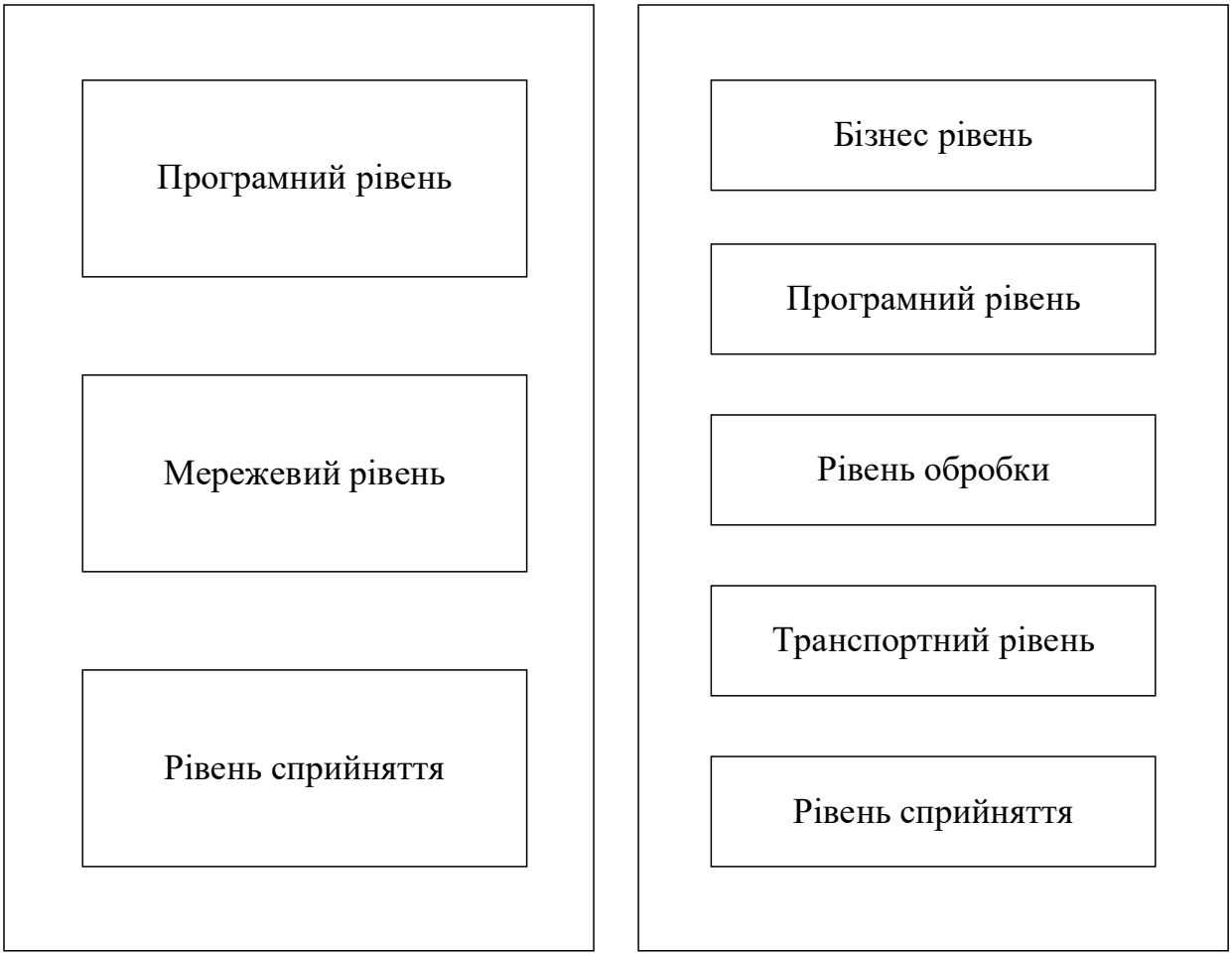
Тришарова архітектура показує головну ідею Інтернету речей, але її недостатньо для більш глибоких досліджень, оскільки дослідження часто зосереджуються на інших аспектах Інтернету речей. Через це ми маємо набагато більше варіантів шарових архітектур. Одним із них є п’ятишарова архітектура, яка додатково включає шар обробки та бізнес-рівень.

До даної архітектури входить п’ять шарів - це рівень сприйняття, транспортування , обробки, програмний та бізнес-рівень (див. Рисунок 1). Роль рівня сприйняття та програмного рівня така ж, як і в архітектурі з трьома шарами. Окреслимо функцію решти трьох шарів:

- *Транспортний рівень* передає дані датчика від рівня сприйняття до рівня обробки і навпаки через такі мережі, як бездротові мережі, 3G, LAN, Bluetooth, RFID та NFC.

- *Шар обробки* також відомий як проміжний рівень. Він зберігає, аналізує та обробляє величезні обсяги даних, що надходять із транспортного рівня. Він може управляти та надавати різноманітний набір послуг нижчим шарам. Він використовує безліч технологій, таких як бази даних, хмарні обчислення та модулі обробки великих даних.

- *Бізнес-рівень* керує всією системою IoT, включаючи додатки, моделі бізнесу.



Тришарова архітектура

П’ятишарова архітектура

Рисунок 1.1 Трьох та п’ятишарові архітектури IoT.

Окрім багатошарової структури, IoT складається з декількох функціональних блоків, які виконують різні функції , такі як зондування, аутентифікація та ідентифікація, контроль та управління. Існує кілька важливих функціональних блоків, відповідальних за операції вводу-виводу даних, вирішення проблем з підключенням, обробку даних, моніторинг аудіо/відео та управління сховищем. Всі ці функціональні блоки, працюючи разом створюють ефективну систему IoT. Хоча існує декілька еталонних архітектур, запропонованих зі своїми технічними специфікаціями, але вони всі ще далекі від стандартної архітектури, яка підходить для глобальної

IoT. Тому, як і раніше, потрібна відповідна архітектура, яка може задовольнити глобальні потреби в Інтернеті речей. Загальна робоча структура системи IoT показана на рисунку 2 .Цей рисунок показує залежність IoT від певних параметрів програми. Шлюзи відіграють важливу роль в функціонуванні IoT, оскільки вони дозволяють зв'язувати сервери та пристрої , пов'язані з декількома програмами.

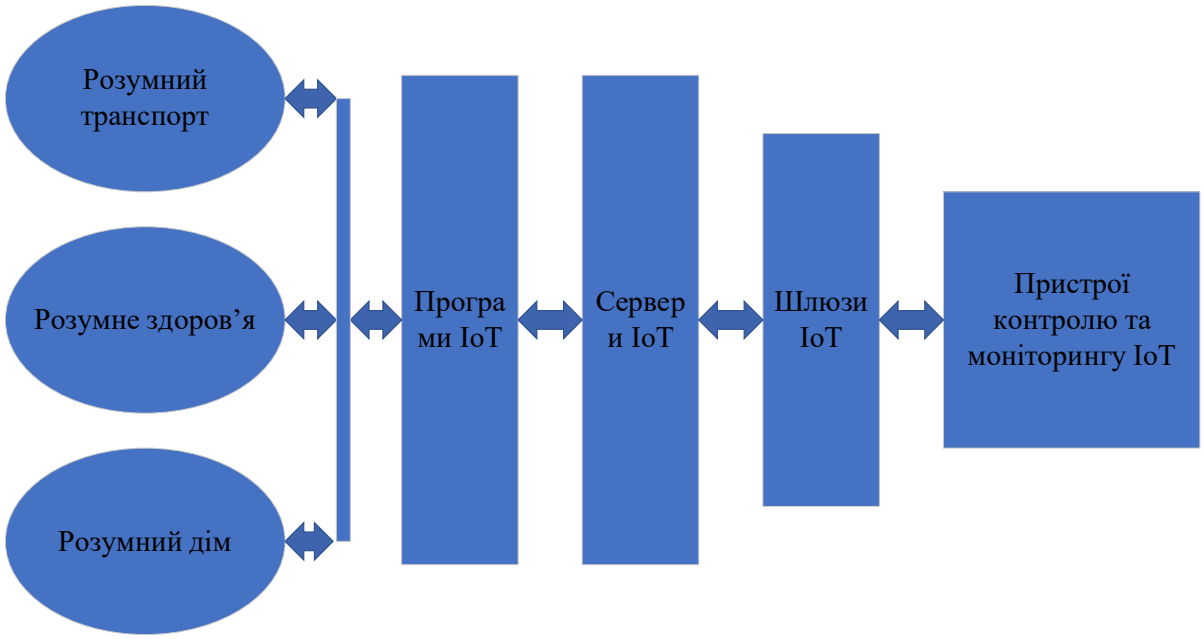


Рис 1.2 Структура системи IoT

Масштабованість, модульність, сумісність та відкритість - це ключові питання при проектуванні ефективної архітектури IoT в гетерогенному середовищі. Архітектура IoT повинна бути розроблена з метою задоволення вимог міждоменної взаємодії, багатосистемної інтеграції з потенціалом простих та масштабованих функціональних можливостей управління, аналітики та зберігання великих даних та зі зручними для користувача програмами. Крім того, архітектура завжди повинна мати можливість розширити функціональність та автоматизацію серед пристроїв IoT в системі.

На першому етапі датчики та виконавчі механізми відіграють головну роль. Реальний світ складається з навколишнього середовища, людей, тварин, електронних гаджетів, інтелектуальних транспортних засобів та будівель

тощо. Датчики виявляють сигнали та потік даних від цих реальних істот і перетворюються на дані, які надалі можуть бути використані для аналізу. Більше того, виконавчі механізми здатні втручатися в реальність, тобто контролювати температуру в приміщенні, сповільнювати швидкість автомобіля, вимикати музику та світло, тощо. Тому перший етап допомагає збирати дані з реального світу, які можуть бути корисними для їх подальшого аналізу. Другий етап відповідає за співпрацю датчиків з виконавчими механізмами, а також зі шлюзами та системами збору даних. На цьому етапі масивний обсяг даних, що генеруються на першому , агрегується та оптимізується структурованим способом, придатним для обробки. Як тільки масивний обсяг даних буде агрегований і структурований, він буде готовий до передачі на третій етап. Крайові обчислення можна визначити як відкриту архітектуру, яка дозволяє використовувати технології IoT та величезну обчислювальну потужність з різних місць у всьому світі. Це дуже найкращий підхід для потокової обробки даних і він добре підходить для систем IoT. На третьому етапі , обчислювальна техніка мають справу з величезним обсягом даних і виконує різні функції, такі як: візуалізація, інтеграція даних з інших джерел, аналіз з використанням методів машинного навчання тощо. Останній етап включає кілька важливих заходів, такі як: поглиблена обробка та аналіз , надсилання зворотного зв'язку для підвищення точності та точності всієї системи. Все на цьому етапі буде виконуватися на хмарному сервері або в центрі обробки даних. Структура великих даних, така як Hadoop та Spark, може бути використана для обробки цих великих потокових даних, а підходи машинного навчання можуть бути використані для розробки кращих моделей прогнозування, які могли б допомогти у більш точній та надійній системі IoT для задоволення потреб сучасного часу.

1.3 Зв’язок клієнт-сервер

Більшість архітектур, запропонованих для IoT, також мають свою архітектуру на стороні сервера. Сервер підключається до всіх взаємопов’язаних компонентів, об’єднує послуги та діє як єдиний пункт обслуговування для користувачів.

Архітектура на стороні сервера зазвичай складається з трьох шарів. Перший - це базовий рівень, що містить базу даних, що зберігає деталі всіх пристроїв, їх атрибути, інформацію про метаінформацію та їх взаємозв'язки. Другий рівень (компонентний рівень) містить код для взаємодії з пристроями, запиту їх стану та використання їх підмножини для надання послуг. Найвищий рівень - це прикладний рівень, який надає послуги користувачам.

На стороні пристрою (об'єкта) існує два шари. Перший - це об’єктний рівень, який дозволяє пристрою підключатися до інших пристроїв, розмовляти з ними (за допомогою стандартизованих протоколів) та обмінюватися інформацією. Об'єктний рівень передає інформацію соціальному шару. Соціальний рівень керує виконанням програм користувачів, виконує запити та взаємодіє з рівнем додатків на сервері.

1.4 Основні компоненти розумних пристроїв

При дослідженні та вивченні інтернету речей, ми представляємо пристрої, як ботів, які можуть встановлювати зв’язки між собою та змінювати їх з часом. Це дозволить пристроям безперешкодно взаємодіяти між собою та виконувати складні завдання.

Щоб така модель працювала, нам потрібно мати багато компонентів, які злагоджено працюють між собою.

Розглянемо деякі основні компоненти такої системи.

- *Ідентифікатор*: для злагодженої роботи системи нам потрібен унікальний метод ідентифікації кожного об’єкта. Ідентифікатор може бути присвоєний об'єкту на основі традиційних параметрів, таких як MAC ID, IPv6 ID, універсальний код товару або інший спеціальний метод.

- *Метаінформація*: поряд з ідентифікатором нам потрібна деяка метаінформація про пристрій, що описує його форму та роботу. Це потрібно для встановлення відповідних взаємозв’язків з пристроєм, а також належного розміщення його у мережі пристроїв IoT.

- *Елементи керування безпекою:* це схоже на налаштування “списку друзів” у Facebook. Власник пристрою може встановити обмеження щодо типів пристроїв, які можуть до нього підключатися. Вони зазвичай називаються елементами керування власником.

- *Виявлення послуг інших пристроїв*: такий тип системи схожий на хмару послуг, де нам потрібно мати спеціальні каталоги, які зберігають інформацію про інші пристрої, що надають певні види послуг. Дуже важливо постійно оновлювати ці каталоги, щоб пристрої могли дізнаватися про інші пристрої в мережі.

- *Управління відносинами*: цей модуль управляє взаємозв’язками з іншими пристроями. У ньому також зберігаються типи пристроїв, з якими певний пристрій повинен підключитися залежно від його завдання. Наприклад, для сенсора світла має сенс встановити взаємозв’язок із датчиком світла

- *Склад послуг*: цей модуль виводить соціальну модель IoT на новий рівень. Кінцевою метою наявності такої системи є надання кращих інтегрованих послуг користувачам. Наприклад, якщо на кондиціонері є жатчик потужності, і цей пристрій встановлює взаємозв’язок з аналітичним механізмом, тоді цей механізм може надати багато даних про схеми використання кондиціонера. Якщо соціальна модель є більш експансивною, а пристроїв набагато більше, то можна порівняти дані зі схемами використання інших користувачів і отримати ще більш значущі дані.

1.5 Протоколи та технології IoT

Протокол - це набір правил, яких слід дотримуватися.

Протокол зв’язку - фраза використовувана для опису набору правил якими користується комунікаційне обладнання при відправці даних на інші пристрої. Якщо пристрої які намагаються обмінюватися даними один з одним використовують різні набори правил, то пристрої не зрозуміють один одного.

Протоколи IoT є найважливішою частиною стеку технологій IoT - без них апаратне забезпечення стало б марним, оскільки протоколи IoT дозволяють йому обмінюватися даними структурованим способом. З переданих частин даних можна ортимати корисну інформацію для кінцевого користувача, і завдяки цьому все розгортання системи стає економічно вигідним, особливо з точки зору управління пристроями IoT .

Говорячи про Інтернет речей , ми завжди думаємо про злагоджену роботу кожного елемента. Взаємодія між датчиками, пристроями, шлюзами, серверами та користувацькими програмами є основною характеристикою, яка робить Інтернет речей таким, яким він є. Але те, що дозволяє всім цим розумним речам говорити та взаємодіяти, - це протоколи IoT, які можна розглядати як мови, які використовує IoT для спілкування.

Протоколи для систем IoT в основному діляться на чотири категорії:

- Для зв'язку на коротких відстанях;
- Для зв'язку на середніх відстанях ;
- Для зв'язку на великих відстанях;
- Для дротового зв'язку,

Технологія *зв'язку на коротких відстанях* має низьку затримку та є дуже надійною. Вона є достатньо безпечною та підтримує найкращу сумісність. Через малу дальність використання, вона долає дуже мало перешкод, навіть в екстремальних погодних умовах. Це є причиною її

широкого використання для зв'язку між пристроями IoT. До протоколів цієї категорії відносяться:

Z-Wave, ZigBee, Wi-Fi, Wi-Fi direct, RFID, Bluetooth, Li-Fi, NFC.

Зв'язок середнього діапазону - це одне і те ж , що з’єднання за допомогою бездротового зв'язку локальних мереж (WLAN). Для потреб систем IoT були створені два нових протоколи для підтримки зв’язку середнього діапазону при зменшенні споживання електроенергії.

Wi-Fi HaLow (IEEE 802.11ah) розширює сучасний діапазон Wi-Fi майже вдвічі, одночасно забезпечуючи краще просочення сигналу через перешкоди.

LTE-Advanced - це мобільний стандарт, що підтримує 4G швидкість передачі даних мережами мобільних операторів.

Комунікація на великих відстанях, майбутній протокл LoRaWAN , який зараз знаходиться в розробці, забезпечить високу енергоефективність при значних комунікативних можливостях. Протокол побудований на базі LoRa компанією LoRa Alliance. Це дозволить швидко налаштувати мережі IoT в будь-якому місці. Мережі можуть бути як загальнодоступними, так і приватними. Сучасні протоколи мобільної педачі даних призначені для високоякісної передачі даних, але їх використання коштує дорого, і вони не є енергоефективними.

Дротове зв’язок - це термін, який використовується при передачі даних по дротовій мережі. Ethernet – це основний протокол канального рівня в стеці TCP / IP.

1.6 Основні операційні системи пристроїв IoT

Сучасні розумні девайси розвиваються з кожним днем. Це призводить до того що вони стають все більш складними. Зростаюча складність робить роботу на бінарному рівні досить складною, тому операційні системи IoT досить потужні. Існує кілька ОС для пристроїв інтернету речей:

- ARM-mbed - зазвичай використовується у переносних пристроях

- Zephyr - в основному використовується в малопотужних пристроях;

- Ubuntu Core - широко використовується в безпілотних літальних апаратах, роботах;

- Apache Mynewt - в основному працює на пристроях що використовують BLE протокол ;

- RIOT - є безкоштовним і відкритим вихідним кодом і дуже простим в налаштуванні;

- RealSense - використовується в різних датчиках і камерах;

- Nucleus RTOS - забезпечує використання і адміністрування БД, підтримує USB, мультимедіа, та має базову підтримку графічного інтерфейсу користувача. Використовується в промислових приладах аерокосмічної промисловості;

- Contiki - використовується в малопотужних бездротових пристроях таких як вуличні системи освітлювання, звукові системи та системи моніторингу;

- Integrity RTOS – дуже потужна ОС, призначена для багатоядерних комп'ютерів. Широко використовується у військовій справі

1.7 Сфери життя, де активно розвивається IoT

1.7.1 Економіка, охорона навколишнього середовища та охорона здоров'я

IoT функціонує для забезпечення суспільних та фінансових потреб людей та розвитку суспільства. Він допомагає оптимізувати процеси в деяких державних установах. IoT може допопогти в економічному розвитку, покращенню якості води, добробуті громадян, індустріалізації, тощо. В цілому IoT наполегливо працює над досягненням соціальних, медичних та економічних цілей. Екологічна ефективність – важливе зав. Розробники IoT повинні турбуватися про вплив на навколишнє середовище систем та

пристроїв IoT для подолання їх негативного впливу. Споживання енергії пристроями IoT є однією з проблем, пов’язаних із впливом на навколишнє середовище. Споживання енергії зростає високими темпами через найсучасніші пристрої, що ботребує система та послуги з підтримкою Інтернету. Ця область потребує глибших досліджень для розробки високоякісних матеріалів, щоб створити нові пристрої IoT з меншим рівнем споживання енергії. Крім того, можна використовувати зелені технології для створення ефективних енергоефективних пристроїв для майбутнього використання. Це не тільки екологічно, але й корисно для здоров’я людини. Дослідники та інженери беруть участь у розробці високоефективних пристроїв IoT для моніторингу проблем зі здоров'ям, таких як діабет, ожиріння або депресія . Питання, що стосуються навколишнього середовища, енергетики та охорони здоров'я, розглядаються в ряді досліджень.

1.7.2 Розумне місто та транспорт

IoT перетворює традиційну громадянську структуру суспільства на високотехнологічну структуру з можлвістю впровадження концепції розумного міста, розумного будинку та розумних транспортних засобів. Швидке вдосконалення відбувається за допомогою допоміжних технологій, таких як машинне навчання, обробка різних мов, щоб зрозуміти наші потреби та використання технологій кожним з нас вдома. Різні технології, такі як хмарні сервери, бездротові сенсорні мережі, які повинні використовуватися із серверами IoT, щоб забезпечити ефективне функціонування розумного міста.

Іншим важливим питанням є екологічний вплив розумного міста. Тому енергоефективні та зелені технології також слід враховувати при проектуванні та плануванні інтелектуальної міської інфраструктури. Крім того, інтелектуальні пристрої, вбудовані в нещодавно запущені транспортні

засоби, здатні виявляти затори на дорозі і, отже, можуть запропонувати водію оптимальний альтернативний маршрут. Це може допомогти зменшити затори в місті. Крім того, розумні пристрої з оптимальною вартістю повинні бути розроблені та включені у всі транспортні засоби для моніторингу активності двигуна. IoT також дуже ефективний для підтримки технічного стану автомобіля. Розумні автомобілі мають можливість спілкуватися з іншими транспортними засобами за допомогою інтелектуальних датчиків. Це зробило б рух транспорту більш плавним, ніж автомобілі, керовані людиною, які їздили в режимі зупинки та руху. Ця процедура потребуватиме часу для впровадження у всьому світі. До тих пір пристрої IoT можуть допомогти, передбачуючи затори на дорозі і можуть вживати відповідних дій щоб їх уникнути. Тому компанія-виробник транспортних засобів повинна включати пристрої IoT в свої транспортні засоби, щоб мати перевагу на ринку.

1.7.3 Автоматизація сільського господарства та промисловості

За прогнозами, зростаюче населення у світі досягне приблизно 10 мільярдів до 2050 року. Сільське господарство відіграє важливу роль у нашому житті. Для того, щоб прогодувати таку велику кількість населення, нам слід розвивати сучасні підходи ведення сільського господарства. Отже, існує потреба у поєднанні сільського господарства з технологіями, щоб виробництво могло ефективно вдосконалюватися. Тепличні технології - один із можливих підходів у цьому напрямку. Воно забезпечує контроль параметрів навколишнього середовища з метою поліпшення ефективності виробництва. Однак ручне управління цією технологією є менш ефективним та потребує ручних зусиль та витрат, а це призводить до втрат енергії та зменшення ефективності виробництва. . IoT пропонує рішення, що автоматизують технологічні процеси на виробництві, а також допомагають в управлінні запасами, контролі якості, логістиці та оптимізації та управлінні ланцюгами поставок.

2 ДОСЛІДЖЕННЯ ОСНОВНИХ ЗАГРОЗ КІБЕРБЕЗПЕЦІ В ІОТ

2.1 Визначення вразливостей пристроїв IoT

Вразливість - це слабкість, яка дозволяє зловмиснику виконувати несанкціоновані дії (наприклад, доступ до даних без належних привілеїв . В цій частині роботи будуть розглянуті різні вразливі місця, такі як жорстко закодовані паролі за замовчуванням, недоліки введення команд, порти, що відкриваються без потреби, та ін.

1) Паролі за замовчуванням

Вбудовані паролі вважаються однією з найпоширеніших вразливостей пристроїв IoT. Вона має місце коли пристрій змінює пароль адміністратора, незалежно від того, які параметри безпеки встановив користувач або якщо пароль був змінений користувачем. Оскільки користувачі зазвичай не переймаються безпекою, вони не заглиблюються в конкретні налаштування, які потрібно змінити, щоб вимкнути цей пароль. Існують приклади надання вбудованих паролів доступу несанкціонованим користувачам, а також існують пристрої, що надають доступ кожному, хто входить обліковий запис адміністратора , не вимагаючи введення пароля.

2) Слабкі паролі

Поряд із вбудованими паролями, слабкі паролі є ще однією поширеною проблемою для пристроїв IoT. При встановленні кодового слова для своїх пристроїв, користувачі часто обирають простий пароль, або навіть відмовляються від використання пароля для простішого користування. Це полегшує життя зловмисникам, тому що такий пароль можна легко зламати шляхом підбирання. Крім того, користувачі повторно використовують старі паролі або вносять незначні зміни до них. Якщо один із облікових записів користувача зламали, це піддає ризику його інші облікові записи та дані, оскільки всі вони мають схожий, якщо не однаковий пароль.

3) Вразливості введення команд

Зловмисники використовують цей метод для доступу до до даних, до яких вони не повинні мати доступ при роботі з консолью. Це дозволяє їм отримати доступ, або змінити підключені бази даних або навіть операційну систему, на якій працює пристрій. Поширеними прикладами цього є введення SQL та JavaScript команд , де зловмисник може запускати шкідливий код, щоб змінити вміст бази даних . Оскільки пристрої IoT зазвичай працюють на різних операційних системах Linux, вони вразливі до таких команд. Існує багато різних способів, які може здійснити зловмисник після введення такої команди. Вони можуть змінювати інформацію, яку законний користувач бачить, коли користується пристроєм та може перешкодити йому працювати належним чином, і вони навіть можуть змусити пристрій додавати їх як справжнього користувача, щоб полегшити доступ до нього та всієї мережі. Цей метод також може бути використаний при створенні ботнетів.

4) Відкриті порти

Коли інформація передається через мережу, вона використовує певні типи портів залежно від типу задіяних даних. За замовчуванням усі порти відкриті, щоб забезпечити передачу даних без будь-яких проблем. Однак користувачі можуть закрити будь-які порти, що не використовуються, щоб зменшити поверхню атаки для зловмисників. Брандмауери часто дають можливість закрити більшість портів, які зазвичай не використовуються, але користувачі можуть вносити зміни в кожному конкретному випадку, щоб досягти найкращого рівня безпеки, яку вони можуть отримати після налаштування своїх пристроїв.

5) Відсутність можливості блокувати облікові записи

Під час спроби входу на веб-сайт, у програму чи на пристрій багато служб не мають функції блокування облікового запису, при неправильно введеному паролі. Це дозволяє зловмисникам постійно підставляти будь-які варіанти паролів, пінів або шаблонів, доки їм не вдасться підібрати правильний. Вони можуть або зробити це вручну самостійно, або

налаштувати програмне забезпечення, яке зможе запустити простий підбір, або словникову атаку. Ці атаки намагаються використати всі можливі комбінації пароля, це може тривати скільки завгодно, години, дні, тижні чи навіть довше. Використовується в тандемі з іншими аспектами наприклад, перелік облікових записів, це може значно зменшити час, який займає атака. Це пов’язано лише з необхідністю вгадати ім’я користувача чи пароль, замість того, щоб вгадувати обидва.

6) Незашифровані сервіси

Конфіденційність даних є життєво важливим компонентом інформаційної безпеки. Надійне шифрування зв‘язку та зберігання даних може запобігти отримання розвідувальної інфорації неавторизованим користувачам в разі атаки. Якщо організація вирішить використовувати слабкі стандарти шифрування своїх послуг, вони вразливі до витоку інформації, прослуховування та інших типів розвідки. Це пов’язано з тим, що пристрої надсилають пакети через мережу передачі даних, які можуть бути легко перехопленими та прочитані за допомогою аналізатора пакетів. Звідти зловмисник може просто мовчки відстежувати інформацію, що передається, або маніпулювати нею, при цьому не підробляючи її. Використання більш надійного шифрування, перш за все перешкоджає перехопленню пакетів.

7) Незахищені веб-інтерфейси

Незахищений веб-інтерфейс може існувати у різних формах та може бути шкідливим або просто небезпечним. Приклад. Небезпечний веб-інтерфейс не може використовувати належну пісочницю користувацьких сесій. Це може дозволити зловмиснику використовувати JavaScript ін’єкцію під час атаки міжсайтового скриптингу (XSS). Подібним чином поля введення можуть бути неналежно очищенні, що може дозволити подальше введення команд. Небезпечні інтерфейси можуть використовувати нав'язливі методи для відстеження користувачів та витоків даних. Шкідливі веб-інтерфейси можуть використовуватися зловмисниками, наприклад

підроблена веб-сторінка, що використовується для запуску фішинг-атаки, а також для викрадання логінів та облікових даних.

8) Небезпечні мережеві послуги

Застарілі мережеві протоколи уразливі для безлічі атак. Протоколи, такі як Telnet і блок повідомлень сервера (SMB) містять застарілі контрзаходи або не призначені для забезпечення безпеки через відсутність шифрування за замовчанням. Пристрої можуть бути уразливі використовувати логіни за замовчуванням, такі як login: root з Telnet. SMB також має уразливості, пов'язані з програмами-вимагачами. інфекції, Петя і Нотпетя. Мережеві сервіси стають небезпечними з кількох причин, включаючи переповнення буфера, відкриті порти, експлуатовані служби UDP і інші. Деякі приклади атак, які можуть бути використані проти небезпечних мережевих сервісів, включають: атаки фаззінга, а також використання відкритих портів для отримання доступу до мережі

9) Небезпечні хмарні інтерфейси

Небезпечні з'єднання знижують очікування конфіденційності даних. Небезпечний хмарний інтерфейс може використовувати незахищені протоколи, в яких відсутні функції шифрування Secure Socket Layer (SSL). Небезпечне додаток також може дозволити зловмисникам підслуховувати одночасних користувачів. Це може бути використано для крадіжки даних, включення ненавмисних функцій або порушення конфіденційності користувача. Кілька різних аспектів визначають, чи є хмарний інтерфейс безпечним чи ні, в тому числі відсутність облікового запису перерахування, відсутність блокування облікового запису, незашифровані дані при переміщенні по мережі і інші. Приклади атак: зловмисник може визначити, чи дійсна обліковий запис, спробувавши скинути пароль або використовуючи аналізатор пакетів для пошуку облікових даних через незашифрований трафік.

10) Перерахунок облікових записів

Перерахування облікових записів описує додаток, яке у відповідь на невдалу спробу аутентифікації повертає відповідь. вказує, чи відбулася помилка аутентифікації через невірне ідентифікатора облікового запису або невірного пароля. Це дозволяє зловмиснику визначити дійсні ідентифікатори облікових записів, які розпізнаються додатком. Перерахування аккаунтів полегшує задачу злому пароля, що дозволяє зловмисникам розпізнати дійсний набір ідентифікаторів облікових записів. Серйозність цього може відрізнятися від простого проголошення чогось простого, наприклад «недійсний логін», або чогось більш конкретного, наприклад «цей пароль НЕ пов'язаний з цією адресою електронної пошти »або« С цією адресою електронної пошти не пов'язана жодна обліковий запис ». Повертаючи конкретні запити, такі як ці, він дозволяє зловмисникам звузити коло атак, щоб потрапити до свого облікового запису, знаючи, яка змінна у них правильна. Можливий спосіб запобігти цьому - повідомити користувачеві, що електронний лист було відправлено на вашу обліковий запис, незалежно від того, чи є воно дійсну адресу електронної пошти.

11) Переповнення буфера

Переповнення буфера - це експлойт, що використовує програму, яка очікує входу користувача. Стан переповнення буфера - це особливий ризик для пристроїв IoT через їх обмеженість в пам'яті, мов, на яких вони запрограмовані, і спільності програм. Як правило, пристрої IoT зазвичай потребують малих енергозатрат, що призводить до невеликого обсягу енергоефективної пам'яті. Чим менше розмір буфера, тим легше його переповнити, що робить IoT центром атак такого типу. Хоча компанії все краще і краще запобігають цю атаку на нових пристроях, старі пристрої як і раніше мають цю вразливість. Причиною може бути апаратне забезпечення, на якому вони побудовані, або якщо вони були запрограмовані на мовах, які особливо уразливі для атак, таких як С, багато пристроїв, які все ще використовуються сьогодні, можуть переповнити свій буфер.

12) Відсутність авторизації користувача

Авторизація визначає, чи має користувач необхідні привілеї для доступу до ресурсів та виконання дій. Її зазвичай називають Рольовим контролем доступу (RBAC). Механізм RBAC забезпечує спосіб обмеження доступу до системи та виконую діїї по уповноваженню користувачів. У RBAC контроль доступу базується навколо ролей та привілеїв. За відсутності відповідного механізму авторизації, як тільки користувач проходить автентифікацію, користувач може виконувати будь -які дії, навіть якщо вони завдадуть шкоди.

2.2.Аналіз можливих атак на екосистему IoT

Оскільки мережа IoT включає в себе багато пристроїв, таких як датчики, процесори та багато інших, мета забезпечити обмін даними та підключення до інших мереж була успішно виконана. В цій мережі велика кількість підключених пристроїв, тому дані що вони використовують є незахищеними, і це викликає занепокоєння рівнем безпеки таких систем . Безпека IoT це захист інформації, яка обмінюється між різними мережами через пристрої IoT за допомогою технологій IoT. Ці пристрої підключаються один до одного за допомогою інтернету, через це вони мають купу вразливостей, що дозволяють хакерам викрадати дані. Незахищені дані можуть призвести до багатьох проблем і принесуть величезні збитки багатьом галузям промисловості і навіть людям.

На рисунку 2.1 показана архітектура системи IoT із шарами, на яких можуть відбуватися атаки. Як було згадано в попередньому розділі система IoT зазвичай складається з трьох основних рівнів, таких як: сприйняття, мережевий та прикладний(програмний) (див. рис. 1.1). Шар сприйняття є найнижчим шаром звичайної архітектури IoT. Цей шар складається з пристроїв, датчиків і контролерів. Основним завданням цього шару є збір цінної інформації з сенсорів та датчиків.

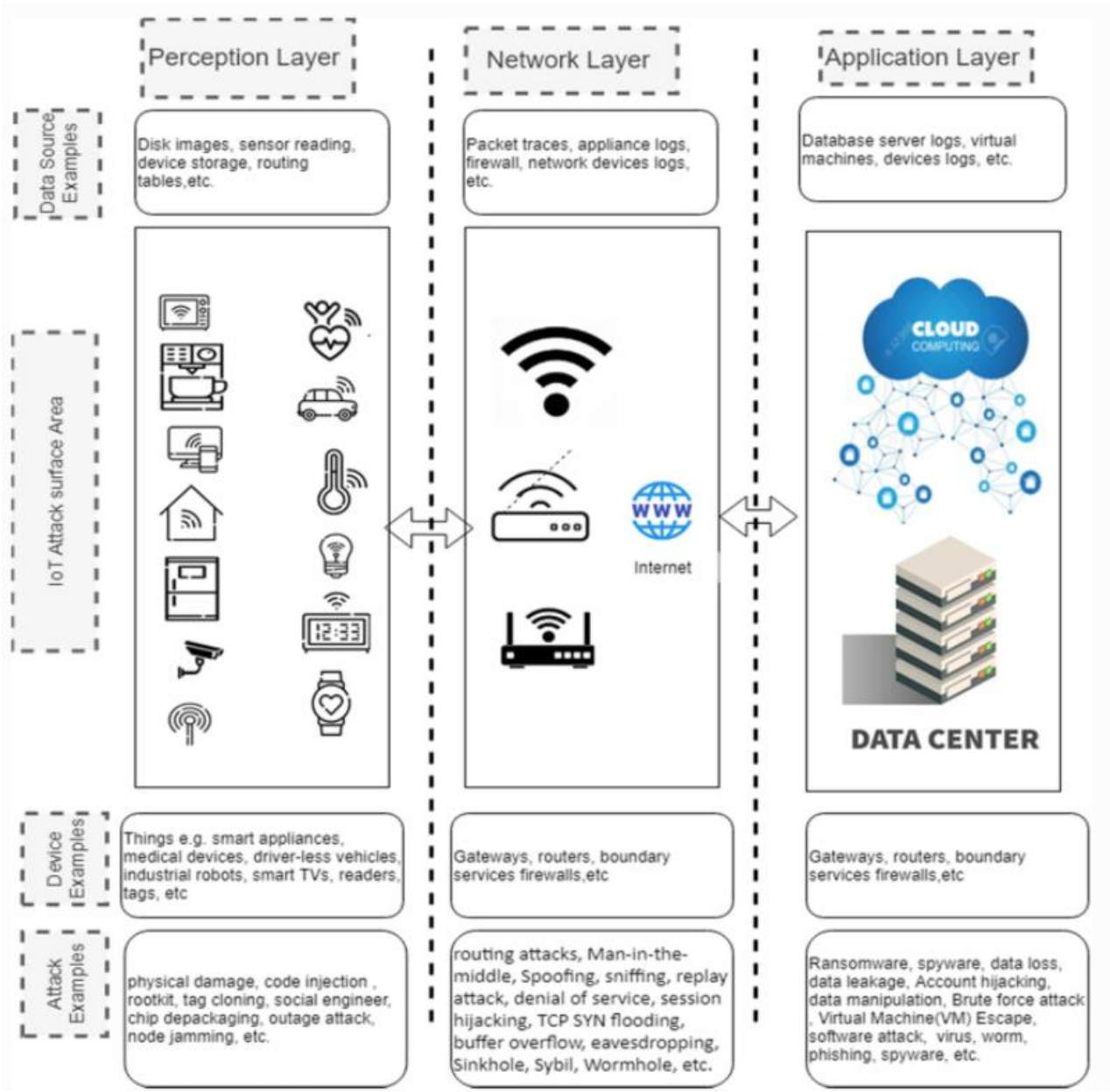


Рис 2.1 Шари IoT на яких можуть відбуватись атаки.

IoT привернув увагу людей та організацій з багатьох секторів, надавши їм надзвичайні переваги. Разом з надзвичайним зростанням цієї технології стали помітними деякі проблеми безпеки, через які відбувалися атаки на IoT, заважаючи людям безпечно використовувати цю технологію. До пристроїв IoT можна отримати доступ з будь-якого місця в довіреній мережі. Отже, існує велика кількість типів зловмисних атак у мережі IoT. Питання безпеки та конфіденційності повинні бути належним чином розглянуті в IoT, щоб захистити його від зловмисних атак.

На мережевому рівні IoT знаходяться різноманітні мережі, такі як WSN, WLAN тощо. Ці мережі допомагають датчикам в IoT обмінюватися інформацією. Шлюз може спростити зв'язок декількох датчиків по мережі. Мережевий рівень забезпечує успішну передачу даних, тоді як прикладний рівень є найвищим рівнем, який обробляє дані для візуалізації.

Більшість зловмисників націлені на мережу пристроїв та обладнання IoT, а не на один ПК. IoT має взаємозв'язок між різними пристроями та обладнанням. Основні причини, чому IoT є ціллю зловмисного програмного забезпечення нведені нижче:

- Усі пристрої та обладнання в Інтернеті речей є постійно увімкненими, тому зловмисники можуть легко зламати їх в будь-який момент часу;
- Пристрої та обладнання, взаємопов’язані один з одним, вони завжди підключені, і зловмисники можуть отримати доступ до всієї мережі з одного пристрою;
- У більшості випадків належні заходи безпеки важче організувати на великій кількості взаємопов’язаних пристроїв, ніж на одному ПК;
- Відсутність належного шифрування даних у взаємопов’язаних пристроях та слабкі паролі;

На рисунку 2.2 показані найпоширеніші типи кібератак, які націлені на мережі IoT. Нижче описано типи атак, а також те, як кожна з них вплине на мережу IoT.

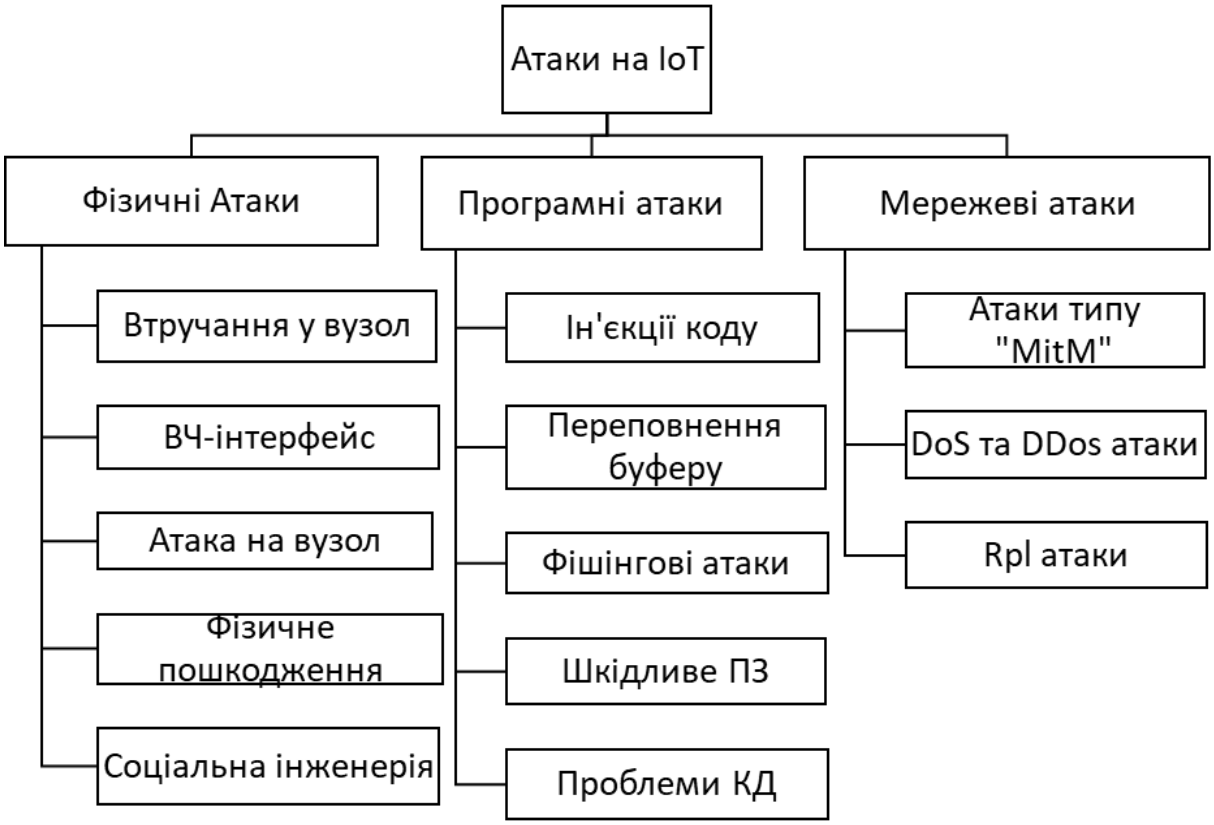


Рисунок 2.2 Типи кібератак на IoT

2.2.1 Атаки на фізичному рівні

Атаки , що базуються на прихованих аспектах пристроїв та обладнання. Ці атаки можуть взяти під контроль пристрій, підробляючи обладнання. Фізичні атаки IoT запускаються, коли зловмисник знаходиться близько до мережі або пристрою IoT. Серед основних загроз на фізичному рівні є:

1) Втручання у вузол

Втручання в вузли стосується злому системи для пошуку секретних ключів для дешифрування зашифрованих даних.

2) ВЧ нав’язування

Радіочастота (RF), яка використовується для бездротового зв'язку між IoT може бути зашумлена. Ця бездротова технологія передачі даних між пристроями вразлива до кількох атак, які можуть легко пошкодити пристрої IoT.

3) Перевантаження вузлів

Це тип DoS-атаки, в якій зловмисник передає сигнал високого діапазону, щоб перешкоджати нормальній передачі даних. Зловмисний вузол у мережі сенсорів передає сигнал, який має подібний набір частот, до того, яким користуються датчики. Ця атака зупиняє роботу датчиків, вони не можуть передавати та приймати дані, створюючи шум в мережі IoT і роблячи послуги недоступними.

4) Копіювання пристроїв

Кіберзлочинець може отримати повний контроль над датчиками. Через розміщення пристроїв IoT в різних місцях мітки сприйнятливі до фізичних атак. Кіберзлочинець міг просто взяти ці мітки та зробити їх копію, для подальшого використання системою RFID.

5) Фізичне пошкодження

Зловмисник може фізично брати участь в атаці, щоб змінити дані або викрасти конфіденційну інформацію.

6) Атаки соціальної інженерії

Зловмисник може використовувати методи соціальної інженерії для незаконного доступу до системи та таємного встановлення шкідливого програмного забезпечення. Пристрої IoT, зокрема переносні пристрої, збирають величезні обсяги інформації, що ідентифікує особу (PII).

2.2.2 Атаки програмного рівня

1) Шкідливе програмне забезпечення

Шкідливе програмне забезпечення - це будь-яке зловмисне програмне забезпечення, призначене для заподіяння шкоди архітектурі IoT. Існує велика різноманітність шкідливих програм, включаючи віруси, хробаки, троянські коні, програми-вимагачі, шпигунське та рекламне ПЗ.

2) Фішинг-атака

Зловмисник використовує один з вузлів IoT як пастку. Мета даної атаки - зібрати таку інформацію, як паролі, імена користувачів тощо.

3) Атака сторонніми каналами

Атака бічного каналу порушує криптографію, використовуючи інформацію, розкриту криптографією.

2.2.3 Атаки на мережевому рівні

Передача даних відбувається на мережевому рівні, де існують проблеми безпеки, що можуть призвести до атак. Більшість атак на IoT здійснюються саме на мережу та проводяться з метою пошкодження певних властивостей інформації. Зазвичай це цілеспрямовані атаки, що завдають шкоди доступності IoT або впливають на конфіденційність даних. Деякі з найважливіших загроз на мережевому рівні:

1) Атака "людина посередині" (MITM)

Бездротовому зв’язку датчиків може загрожувати людина, яка атакує посеред, вона може загрожувати конфіденційності, цілісності та доступності даних, що передаються. Бездротові атаки можуть бути проведень шляхом злому шифрування, зловмисними бездротовими пристроями, підслуховуванням, підміною MAC, перехопленням пакетів тощо. Атака MITM відбувається, коли зловмисник без дозволу користувача автентифікації змінює зв'язок між двома сторонами, які вважають, що зв’язок

є захищеним. Вона схожа до нападу підслуховування, коли зловмисник може втиснутись у спілкування двох сторін. Існують різні типи атак MITM - це перехоплення електронної пошти, прослуховування WIFI, викрадення сеансів, підробка DNS та підробка IP. Наприклад, зловмисник може вкласти мережеве шпигунське програмне забезпечення (sniffer) на комп'ютер або сервер, щоб здійснити шпигунську атаку та захопити пакет під час передачі. Більше того, будь-який пристрій в мережі між пристроєм що передає і приймальним пристроєм є вразливим для зловмисників,

2) Атака типу відмова в обслуговуванні (DoS)

Атака відмови в обслуговуванні перешкоджає постійній доступності послуг, що надаються в системі. Законні користувачі системи позбавлені можливості використовувати ресурси. Якщо цю атаку запускає велика кількість шкідливих вузлів, то вона називається розподіленою відмовою в обслуговуванні (DDoS). DOS-атака обійдеться жертві не втратою інформації, а значними витратами часу та грошей. DoS-атаки можуть впливати на мережеві ресурси, пропускну здатність та роботу процесора. Оскільки пристрої та обладнання IoT підключені до Інтернету протягом 24 годин і завжди включені, існує висока ймовірність нападу на мережу IoT. Корисне навантаження зловмисного програмного забезпечення можна надіслати в будь-який час в домашній або офісній мережі IoT. Наприклад, `` Mirai" - це ботнет, який здійснив атаку з розподіленою відмовою в обслуговуванні (DDoS), що залишило більшу частину мережі неприступною.

3) Розподілена відмова в обслуговуванні (DDoS)

Під час DDoS-атаки зловмисник на короткий час захоплює кілька пристроїв IoT в певну групу, відому як бот-мережа, а потім робить синхронізовані запити на сервер або масив серверів для ортимання певної послуги, таким чином перевантажуючи сервер, що змушують його не обслуговувати справжні запити від кінцевих користувачів. Зазвичай це трапляється, коли усіма пристроями маніпулюють, а повідомлення

перевантажують пристрої IoT. Даний тип атаки в основному використовується для створення заторів на пристроях.

2.3 Процес атаки на IoT

Разом зі знанням інструментів якими володіють хакери, також дуже важливо знати про можливі переваги цих інструментів, в яких більшість хакерів і зацікавлені. В більшості випадків мова йде про отримання інформації, програмами-виагачами, та діями які завдають шкоду і можуть спричинити порушення безпеки, пошкодження, чи будь-який інший не бажаний ефект системі.

Точки поверення (півот точки) також знаходяться в центрі уваги хакерів, тому що вони часто недостатньо добре відслідковуються, і потрапивши в мережу хакери можуть діяти без ризику бути спійманими.

Останнім, але не менш важливим, є шпигунство через пристрої моніторингу, такі як камери безпеки, мікрофони тощо. Коли мова йде про злам системи, тут немає правил чи кращих практик. Взлом означає мислити поза межами, нестандартно і потребує хороших знань та креативного підходу.

Саме тому так важко запобігти злому систем. Атаки зазвичай проходять у різних фазах:

Етап 1*:* Розвідка - дослідження цілі, до того, як вона буде уражена. Вона може бути пасивною без прямої взаємодії з ціллю. Онлайн-ресурси, наукові праці, веб-сайти, наявна документація, соціальна інженерія тощо, використовуються на цьому етапі.

Етап 2*:* Пошук вразливих місць - коли зібрано достатньо інформації розвідка за допомогою певних інструментів, таких як порти або інші сканери вразливості, хакери намагаються зібрати інформацію про особливості системи та знайти інші вразливі місця. Будь-яка вразливість дозволяє створити новий вектор атаки.

Етап 3: Атака - коли вразливі місця виявленні, тоді атаки спрямовуються проти їх. Існує досить багато варіантів атак. Новою специфічною атакою IoT є прокатні коди (rolling codes), які широко використовуються в галузі автоматизації (замки на ключі тощо). Це робиться з допомогою різних інструментів, таких як RF-Crack, глушителів сигналу, сканерів, чи повторних програвачів, тощо.

Етап 4: Отримання доступу - задіяні шляхи проходять через недостатньо захищені мережі, вразливі чи переписані ОС або програми.

Етап 5: Забезпечення постійного доступу, утримання анонімності - підтримка доступу так само важлива як отримання доступу. Отже, хакери докладають багато зусиль, щоб досягти повного контролю над (PWN) системи. Очищення слідів шляхом видалення журналів дозволяє довше приховати успішну атаку Використання ураженої системи як точки повернення дозволяє отримати доступ до інших ресурсів, переписуючи прошивку з вбудованими бекдорами, а також дозволяє необмежене використання пристрою або системних ресурсів. Мабуть, найважливіший крок, який робиться, коли система уражена - це її зміцнення так щоб ускладнити інші можливі атаки інших хакерів.

3 СПОСОБИ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ В IOT

3.1 Визначення засобів контролю безпеки для відомих вразливостей IoT

Відповідно до багаторівневої архітекти IoT-систем, можна виділити наступні 3 ділянки, на яких необхідно забезпечити інформаційну безпеку:

- Smart-пристрої - «розумні» датчики, сенсори і інші прилади, які збирають інформацію з обладнання і відправляють її в хмару, передаючи назад керуючі сигнали зі змінами стану речей;
- Мережеві шлюзи і канали передачі даних (провідні та безпровідні протоколи);
- Програмні IoT-платформи – хмарні сервіси зберігання та обробки інформації.

Для всіх цих компонентів зокрема і в цілому для IoT-системи актуальні наступні заходи забезпечення кібербезпеки:

Організаційні заходи:

- створення і впровадження єдиної політики інформаційної безпеки підприємства з урахуванням всіх додатків і систем Internet of Things;
- розробка правил безпечного використання IoT-приладів і мереж;
- вдосконалення законодавчого забезпечення недоторканності приватного життя і промислової таємниці;
- державна і приватна стандартизація і сертифікація пристроїв, каналів передачі даних, сховищ інформації та прикладного ПО з обробки та аналізу даних;

Технічні інструменти захисту даних від витоків, втрат і перехоплення управління:

- шифрування і інші криптографічні методи, в т.ч. персоналізація IoT-пристроїв з використанням унікальних ідентифікаторів ID, MAC-адрес,

ключів і сертифікатів, що забезпечують досить високий рівень кібербезпеки без додаткових витрат ;

- гнучкі політики управління доступом з багатофакторним авторизації;
- резервування, організація захищеного периметра і інші засоби інформаційної безпеки для інформації.

3.1.2 Особливості захисту при передачі даних між пристроями IoT та сервером

Інтернет речей це комплексна система триланкової архітектури: інформація з «розумного» приладу або датчиків / сенсорів і інших кінцевих пристроїв, встановлених на технологічному обладнанні, не відразу потрапляє в хмару IoT-платформи для інтелектуального аналізу даних засобами Big Data і Machine Learning.

Кожен компонент цієї складної системи є потенційно вразливим. У той час, як хакери винаходять все нові способи злому smart-пристроїв і хмарних кластерів, фахівці з кібербезпеки роблять всілякі заходи щодо поліпшення захисту Internet of Things: від сертифікації існуючих до розробки нових технологій.

Зокрема, велика увага приділяється каналах передачі даних - інформаційна безпека цього рівня забезпечується багатьма і мережевими протоколами. Поки дані з кінцевого пристрою досягнуть хмарної IoT-платформи, вони передаються через безліч мереж різного типу, що належать різним провайдерам. При цьому інформація розшифровується і знову зашифрована шлюзами безпеки на кожній дільниці, що ускладнює забезпечення комплексної безпеки системи Internet of Things. Таким чином, загальна безпека всієї IoT-системи, в тому числі, залежить від провайдерів і виробників обладнання, що є потенційною вразливістю.

Запобігти можливим перехоплення малих даних на шляху від smart-пристрої до серверу допоможуть наступні додаткові заходи безпеки :

- взаємна аутентифікація пристрою на сервері і сервера на пристрої;
- просте і автоматизоване виділення ресурсів кінцевому пристрою в віддаленому додатку;
- механізми відкликання IoT-пристрої через віддалену програму;
- забезпечення безпечного обміну даними з дотриманням цілісності і конфіденційності (шифрування) повідомлення шляхом використання захищених протоколів, наприклад, IPsec, WPA 802.11, 802.15.4 або 6LoWPAN.
- Перевірка при підключенні нового приладу. Оскільки за замовчуванням усі компоненти системи IoT довіряють один одному, навіть якщо новий прилад, що підключається ніколи раніше не був підключений до даної мережі. Це є серйозною проблемою, адже зловмисник може впровадити свій пристрій в систему, а якщо хоч один з них контролюється зловмисником, то він може легко взяти під контроль всю систему
- Налаштування портів пристрою. Щоразу, коли зловмисники підключаються, вони перевіряють систему на наявність вразливостей. Сюди входять будь-які порти, які є відкритими і можуть надати їм шлях проникнення в мережу. Кожного разу, коли в мережі відкриваються непотрібні порти, мережі стає набагато вразливішою. Мережа із закритими портами дозволяє зменшувати поверхню атаки, тим самим зменшуючи загрозу доступу зловмисника до мережі
- Заходи фізичної безпеки, такі як дверні замки, охорона, картки контролю доступу та системи пожежогасіння фізично захищають пристрої. Це також слід контролювати за допомогою таких пристроїв, як камери спостереження, детектори тепла та датчики виявлення вторгнень. Це гарантує, що пристрої IoT не будуть знищені або ще гірше, вкрадені. Якщо зловмисник отримає фізичний доступ до пристрою, він отримує можливість робити з ним все, що завгодно, незалежно від будь-яких інших заходів протидії, які можуть бути працювати. в

3.2 Криптографічні методи забезпечення безпеки даних в IoT

Криптографія - це метод маскування даних таким чином, щоб заважати третій особі / зловмиснику читати та модифікувати дані, в той час коли вони передаються між відправником та одержувачем повідомлень. Вона відіграє вирішальну роль у захисті інформації в загальнодоступних мереж. Криптографія - це математична функція, що застосовується для шифрування та дешифрування повідомлень. Методи шифрування можна розділити на дві категорії:

- Алгоритми симетричного шифрування також відомі як приватна криптографія;
- Криптографія за допомогою відкритого ключа, яка є асиметричною технікою шифрування.

Симетричне шифрування – це шифрування єдиним ключем, що використовується для шифрування та дешифрування, такими алгоритмами є DES, IDEA, Blowfish, RC4, RC5, RC2, 3DES та AES.

В асиметричному шифруванні використовуються два різні ключі для процесу шифрування та дешифрування, прикладом такого алгоритму є RSA, ElGamal та шифрування за допомогою еліптичної кривої(ECC).

Однак часто вважається, що дані, що передаються, є менш захищеними, що означає коли дані активно передаються по мережі, ефективна схема захисту даних є обов’язковою. Найкращий варіант - легкий метод шифрування, який може впоратися з обмеження ресурсів і виключає реалізацію повного шифрування на пристроях IoT, які вимагають безперебійного живлення, мають невеликий обсяг пам'яті, мають великі розміри програмного коду, що призводить до довшого виконання операцій шифрування.

3.2.1 Гібридне шифрування

В даній роботі, реалізований полегшений гібридний алгоритм шифрування, який забезпечить конфіденційність, цілісність та доступність інформації. Це тристоронній захищений механізм шифрування, в якому ECDH є протоколом узгодження ключів, який забезпечує безперервний обмін ключами криптографії через загальнодоступну мережу, цифровий підпис аутентифікує цифрове повідомлення або документи та алгоритм шифрування шифрує вміст повідомлення, надіслане через небезпечний канал.

3.2.1.1 Короткий огляд алгоритмів ECC, ECDH та AES

Криптографія еліптичної кривої (ECC)

ECC - ц алгоритм шифрування з відкритим ключем, що базується на алгебраїчній структурі еліптичних кривих кінцевого поля. Завдяки властивостям рівняння еліптичної кривої ECC генерує використовувані ключі замість того, щоб застосовувати традиційний метод генерації ключів як результат двох простих чисел. Алгоритм ECC довів, що він забезпечує однакову надійність захисту при менших розмірах ключів, на відміну від традиційних криптосистем, таких як RSA, що призводить до швидшого обчислення та меншого використання.

Алгоритм ECC:

E_c – еліптична крива

P - положення точки на кривій

n - просте число

Генерація ключів:

Це найважливіша частина алгоритму, яка генерує дві різні приватні та публічні ключі . Відкритий ключ одержувача шифрує повідомлення та за яке розшифровується за допомогою закритого ключа.

$X = d \times P$

Де, X = відкритий ключ;

d = вибране число 'd' з діапазоном (від 1 до n-1);

Шифрування: m = повідомлення ‘M’ точка ‘m’ на кривій E_c

K = вибране випадкове ціле число

$C_{i1} + C_{i2} = \text{два зашифрованих тексти}$

$C_{i1} = m + k \times X$

Розшифрування:

$M = C_{i2} - (d \times C_{i2})$

Еліптична крива Діффі-Хеллмана (ECDH)

Еліптична крива Діффі-Хеллмана є альтернативою методе Діффі-Хеллмана для еліптичних кривих. Це ключовий протокол примирення , який встановлює секретний ключ між двома сторонами зв'язку за допомогою еліптичних кривих. Він зазвичай використовується в криптовалюті з еліптичними кривими та має здатність краще запобігати пасивній атаці. Попередні кроки, необхідні для генерації ключів та встановлення спільного секретного ключа:

1) Боб і Франк обирають публічні числа P = 37, G = 10
2) Боб обирає приватним ключем a = 8 ,
   Франк вибирає приватний ключ b = 7
3) Боб і Франк обчислюють публічні значення
   Боб: $y = (10^8 mod 37) = (100000000\ mod 37) = 26$
   Франк: $z = (10^7 mod 37) = (10000000\ mod 37) = 10$
4) Боб та Франк обмінюються публічними ключами
5) Боб отримує відкритий ключ y = 10
   Франк отримує відкритий ключ z = 26
6) Боб і Франк обчислюють симетричні ключі
   Боб: $k_a = y^a\ mod\ p = 100000000\ mod\ 37 = 26$
   Франк: $k_b = z^b\ mod\ p = 8031810176\ mod\ 37 = 26$
7) Спільний секретний ключ – 26

Стандарт попереднього шифрування (AES)

AES - це симетричний блочний шифр, що містить блок довжиною 128 біт. Стандарт містить 3 різні довжини ключів: 128 біт, 192 біти та 256 біт. На етапі шифрування використовується ключ, який перетворює дані в нечитаємий зашифрований текст, потім процес дешифрування використовується той самий ключ для перетворення зашифрованого тексту назад у вхідні дані. Він використовує один ключ, як для шифрування, так і для дешифрування. В алгоритмі AES, кількість повторювань визначається довжиною ключа шифру, тобто 10 повторень для 128-розрядних ключів, 12 для 192-розрядних ключів і 14 циклів для 256-бітних ключів. Чотири етапи трансформації включають: перший крок на основі SubBytes, порядкову перестановку на стадії змішування, етап змішування по колонці та додавання ключа. Кроки шифрування та дешифрування алгоритму:

- SubBytes: цей крок реалізує заміну нелінійної функції, використовуючи метод таблиці пошуку, щоб замінити відповідний байт на інший байт.
- Shift Rows: у матриці кожного стовпця ліворуч від кожного байту циклічне переміщення, переміщення збільшується із збільшенням числа рядків.
- Змішування стовпців: Цей крок поєднує чотири байти кожного стовпця матриці з використанням зворотного лінійного перетворення.
- Додання ключа цику - за допомогою розкладу ключів генерується ключ для кожного циклу, і він поєднується з кожним байтом матриці.

Рис 3.3 Схема алгоритму шифрування AES

3.2.1.2 Запропонована система гібридного шифрування

У запропонованій архітектурі Рис 3.4 схема тристороннього захисту досягається використанням гібридного механізму шифрування, який має переваги симетричного та асиметричного шифру. В алгоритмі симетричного шифрування, ключі криптографії це рядок бітів, що допомагає перетворити відкритий текст у зашифрований для шифрування, і навпаки для дешифрування. Асиметрична криптографія - це будь-яка криптографічна

система, яка використовує пару ключів для шифрування та дешифрування. Ключі неідентичні, ключ, яким ділиться з усіма, є відкритим. Запропоноване рішення не може забезпечити безпечні шляхи комунікації, але це допомагає збільшити криптографічний захист та безпеку загалом.

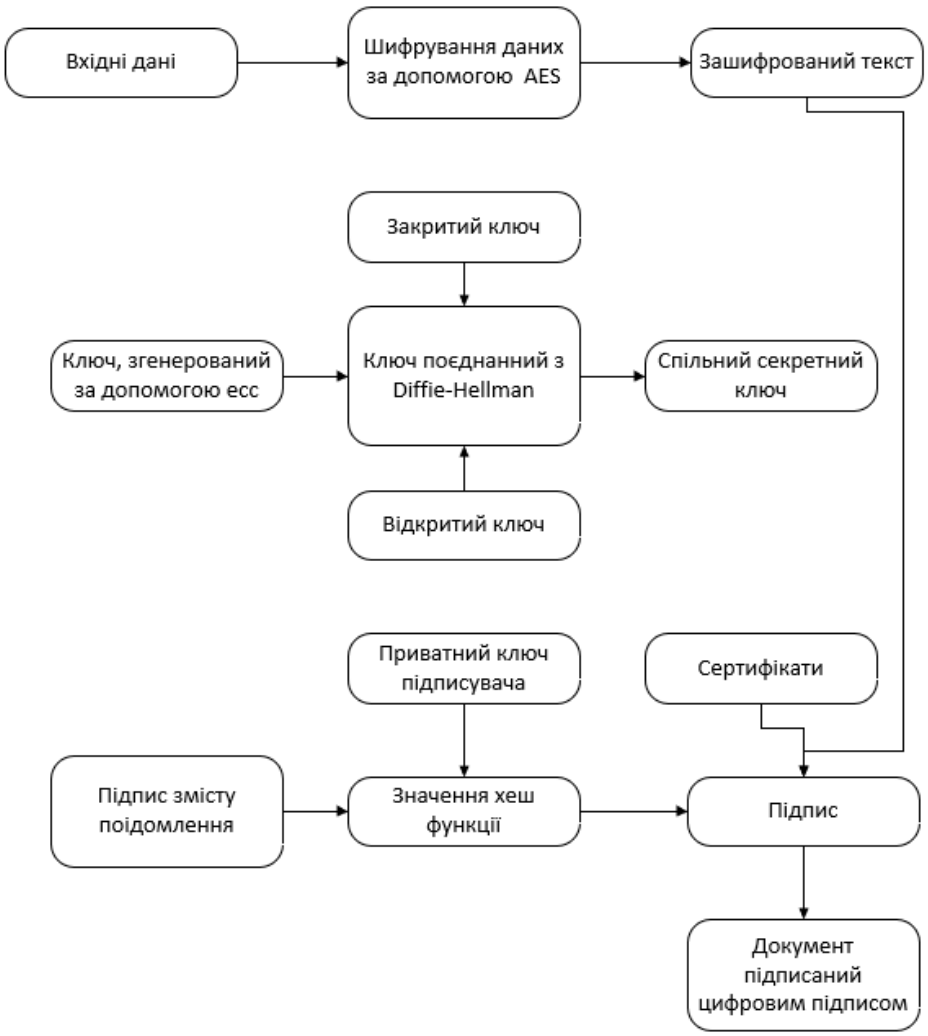


Рис 3.4. Запропонована модель шифрування даних

Метою є ефективне використання техніки шифрування та дешифрування простого тексту та збереження інформації в безпечному вигляді, захищеному від небажаної взаємодії. ECDH це протокол погодження секретного ключа між двома сторонами, які обмінюються даними по публічній мережі. Потім повідомлення аутентифікується за допомогою цифрового підпису. Це підхід, який гарантує, що зміст повідомлення не було змінено під час транспортування через небезпечний канал. Відповідно, алгоритм AES застосовується для шифрування та дешифрування користувацького файлу

даних. Забезпечити ефективні заходи щодо захисту даних при їх передачі та уникати модифікації, все це реалізовано, щоб забезпечити надійне обчислювальне середовище. Запропонована модель проходить два основних етапи, такі як: генерація ключів та їх зміна за допомогою ECDH, а також підпис вмісту повідомлення.

3.2.1.3 Розрахунковий приклад запропонованої моделі.

Етап 1: Генерація та обмін ключами

Припустимо, дві сторони, що спілкуються, Боб і Франк хочуть безпечно обмінюватися інформацією через загальнодоступну мережу, щоб зловмисник міг перехопити їх, але не прочитати.

1) По-перше, Боб і Франк генерують власний відкритий та закритий ключі. Закритий ключ Боба 'bPr', а відкритий 'bPu' = bPrG;

Приватний ключ Франка 'fPr', а відкритий 'fPu' = fPrG; G - цебазова точка, яка породжує підгрупу на еліптичній кривій.

2) Боб і Франк обмінюються своїми відкритими ключами "bPu" і "fPu" по загальнодоступні мережі. Зловмисник може інтерпретувати свої відкриті ключі, але не може скористатись закритими ключами 'bPr' та 'fPr' без вирішення математичної логарифмічної задачі.

3) Боб і Франк знаходять свій секретний ключ S, обчислюючи за допомогою власного приватного ключа та відкритого ключ іншої сторони.

Секретний ключ Боба S = bPr fPu;

Секретний ключ Франка S = fPr bPu;

Боб і Френк тримають один і той же секретний ключ S = bPr fPu = bPr (fPrG) =fPr (bPrG) = fPr bPu

Нижче наведено обчислення відкритих / закритих ключів Боба та Франка, та їх спільний секретний ключ за допомогою еліптичної кривої secp256k1, крива Кобліца від SECG

Крива: secp256k1

Приватний ключ Боба:

0xf8e0ca324ee2f8719a89dfab7c04e6d1bf54a38c35336197e5a957f09031d2d

Відкритий ключ Боба:

(0x525b73fd949508b5f4f66612974069c4e06ebbf334e8d8004e68471fcd4a5b b1,0x9e8324e8d2f025bc627916a149f5ab055e6d20506d0e0ee736a713c6fd58dabf)

Приватний ключ Франка:

0x75cbb6d48faa417342b6873a53d29227fddbf56d5a08583460480a3ae504e6

Відкритий ключ Франка:

(0x3c8947fd1adde6fb79bc896cec1b38cc559116486dc8d082f7b7d23c0347ed 8b,0xe3314117c999c0cf800b93bbb33befc7276600d0026f939dff26c26ce5e13c7a)

Спільний секрет:

(0x2850da8d1c823bc3548dfbdbd13af55eabdf520edf5e508a0d9198cf3d93c1e 0,0x2615448ee7bf3ad321bf7550377a9e8ebe88741c307708bb7c6071539f36259f)

Етап 2: Аутентифікація за допомогою цифрового підпису (ECDSA)

Припустимо, Боб підписав повідомлення своїм приватним ключем (bPr) і він хотів засвідчити підпис повідомлення за допомогою відкритого ключа Франк. Цифровий підпис зазвичай не підписує повідомлення, але вона зберігає хеш повідомлення. Він бере хеш-значення з даних, застосовуючи математичну функцію або алгоритм хешування. Це хеш-значення із сертифікатом і є підписом повідомлення.

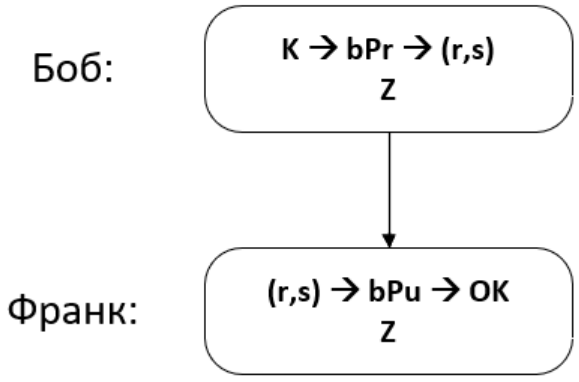


Рисунок 3.5. Підпис та перевірка

Боб використав свій приватний ключ 'bPr' і випадкове ціле число k для підпису хеш-значення. За допомогою відкритого ключа Франка "fPu" Боб підтвердив, що повідомлення підписано правильно і його можна надсилати.

Крива: secp256k1

Приватний ключ:

0x6212a090a08d47e7e17354a57ba2e5c5b338819cebea516744978b3a217a7e

Відкритий ключ:

(0x1cfd0752316066ef46cfcb9a486964fa114c84afd8a9680aa0bff0b067aed2bc
,0x461ec76a5163797fb93f5fe6ab6b5c647b61f1b2e8146bc198cd82a537579872)

Повідомлення:"Хеллоуїн"

Підпис:

(0x8ab4b50a81a5eacfbd345ce21ade99807647a10776f8fe023309305a6a54ad
28,0x41d70340e9736e2ed842d9520dd9d7ad2324b2f6cf064112d35db7a5bca37da)

Перевірка: Збіги підписів

Повідомлення: 'Різдво'

Перевірка: недійсний підпис

Повідомлення: "Хеллоуїн"

Відкритий ключ:

(0x5992a4f9d86a7708ee6a33b456963fcc220c2151cd03f93118651d4e1aad7a
10,0xc423938b8df789df698f751ee696a4b22fdaf86a48ca2412627174ab3b08c2b5)

Перевірка: недійсний підпис

Спочатку повідомлення (Хеллоуїн) було підписано та перевірено підписом. Потім, використовуючи той самий підпис, він намагається перевірити повідомлення (Різдво) і перевірка не вдається. Після цього, використовуючи деякий випадковий відкритий ключ, перевіряє підпис на тій самій кореспонденції прямого повідомлення, перевірка не вдається. Цифровий підпис допомагає автентифікувати цифрове повідомлення, не відмовляючись від надсилання та та не втрачаючи цілісність даних.

3.2.4 Використання спеціальних мікрочіпів для криптографічного захисту

Незважаючи на те, що в більшість IoT-пристроїв вбудовані енергоефективні мікрочіпи криптографічного захисту, вони не захистять від злому або витоку даних, якщо користувач smart-приладу не застосовує їх за призначенням або зловмисник проявляє особливу наполегливість. Наприклад, закриті ключі можна дістати з пам'яті IoT-пристрою, обчислити динамічну зміну струму харчування або навіть по електромагнітному випромінюванню.

Посилити безпеку smart-приладів допоможе установка додаткової захисної електроніки - мініатюрних компонентів, які з'єднують периферійні пристрої з приймаючими микроконтроллерами або мікропроцесорами, і відповідають за персоналізовані сертифікати, безпечне розміщення закритих ключів і управління криптографічними елементами .

Аналогічним чином можна захистити шлюз, який збирає малі дані від smart-пристроїв і передає їх в хмарну IoT-платформу. Для цього до хоста IoT-шлюзу встановлюється додатковий мікрочіп, який забезпечує безперебійне TLS-з'єднання з сервером і виконує завдання забезпечення шлюзу ресурсами по протоколам HTTPS або MQTTS. Відзначимо, що ці прикладні протоколи сімейства TCP / IP також широко використовуються в програмній частині IoT-систем, на стороні хмарної платформи для відправлення / прийому запитів і управління чергами повідомлень в брокерів RabbitMQ, Apache Qpid, Apache ActiveMQ і Apache Kafka.

Найчастіше виробники мікроконтролерів для IoT-пристроїв вбудовують в них спеціалізоване обладнання для реалізації криптографічних алгоритмів. Наприклад, великий виробник мікрочіпів Texas Instruments включає в свої мікросхеми прискорювач алгоритму AES, який реалізує процеси шифрування і дешифрування. Аналогічні апаратні прискорювачі застосовуються для

інших поширених криптографічних функцій, зокрема, для MAC-алгоритму аутентифікації, що використовується для перевірки достовірності.

Найбільш поширеним MAC-алгоритмом є функція SHA (Secure Hash Function), затверджена Національним інститутом стандартів і технологій США. Для її виконання в мікроконтролерах IoT-пристроїв передбачені свої прискорювачі. Наприклад, мікроконтролери Kinetis від компанії Freescale мають співпроцесор для прискорення AES і SHA, який може працювати автономно від центрального процесора і використовувати виділену пам'ять, щоб команди і дані могли бути буферизованного для пристрою криптографічного прискорення.

IoT-електроніка компанії Atmel забезпечує безпеку за допомогою пристрою захисту пам'яті. Воно використовує симетричну аутентифікацію, шифрування даних і MAC-функції, щоб забезпечити безпечне зберігання інформації через стандартний послідовний інтерфейс мікроконтролера. Завдяки додатковим схемами визначення несанкціонованого доступу інформація захищена від атак зовнішніх пристроїв.

Компанія Maxim Integrated виробляє серію мікросхем з апаратною реалізацією стандарту SHA-256 для безпечної передачі даних за допомогою звичайного однопровідного інтерфейсу. Розпізнавання ведучого IoT-пристрої веденим захищає пам'ять від змін, які може внести неідентифікований прилад. Це забезпечує надвисоку надійність, що актуально в IoT-пристроях, коли аутентифікація приладу певного виробника запобігає використання підробленої техніки. У прикладному сенсі це важливо, якщо показники датчиків використовуються, наприклад, для розрахунку плати за комунальні послуги .

3.3 Впровадження систем виявлення та запобігання вторгненям

Для виявлення загроз використовуються мережеві системи виявлення та запобігання вторгненням (IDS/IPS), вони збирають інформацію про вхідний

та вихідний трафік з інтернету (рис. 3.4). Ці системи використовують комбінацію сигнатурних та аномальних методів виявлення. Виявлення на основі сигнатур передбачає порівняння зібраних даних пакетів з файлами сигнатур, які, які завідомо, є шкідливими, в той час як метод виявлення на основі аномалії використовує поведінковий аналіз для відстеження події, відносно базової лінії "нормальної" мережевої активності. Коли в мережі виникає зловмисна активність, IDS/IPS виявляють активність та генерують попередження для сповіщення адміністратора, або блокування вихідної IP-адреси в мережі.

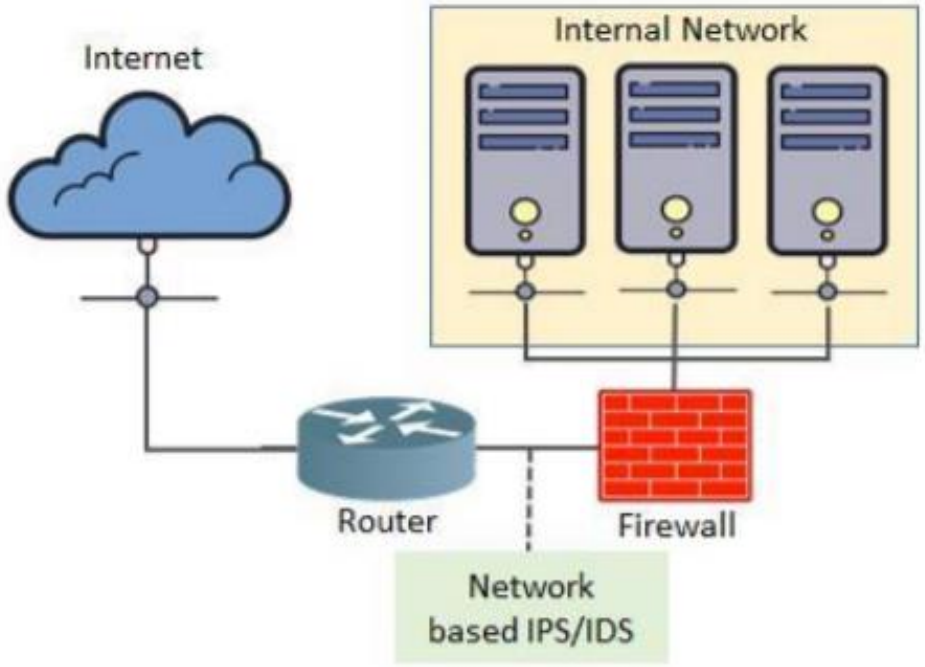


Рисунок 3.4 Схема розміщення IPS/IDS

3.3.1 Огляд систем виявлення та запобігання вторгненням SNORT та SURICATA

Існує кілька двигунів IDS/IPS з відкритим кодом, доступних для автоматизації та спрощення процесу виявлення вторгнень, та Snort - одне з найкращих рішень для малих та малозавантажених мереж. Він був розроблений в 1998 році і нині належить компанії Cisco, яка придбала його у

2013 році . Snort є найпоширенішою IDS / IPS системою у світі за останнє десятиліття. За даними веб-сайту The Snort, цю IDS/IPS завантажено на сьогоднішній день понад 5 мільйонів разів і вона наразі нараховує понад 600 000 зареєстрованих користувачів. Вона має однопотокову архітектуру, яка використовує стек TCP / IP для захоплення та перевірки корисного навантаження мережевих пакетів. Однак остання версія Snort 3.0 додала можливість використовувати потоків обробки пакетів для подолання обмежень однопотокової архітектури в попередніх версіях. Вона використовує обидва методи: сигнатури (SIDS) та аномалії (AIDS) для виявлення трафіку.

Suricata IDS/IPS є порівняно новою IDS/IPS порівняно із Snort. Вона була розроблена у 2010 році Фондом відкритої інформаційної безпеки (OISF) для того щоб задовільнити вимоги сучасної інфраструктури. Ця IDS/IPS використовує багатопоточність для прискорення аналізу мережевого трафіку та подолання обчислювальних обмеженнь однопотокової архітектури. Як Snort, Suricata базується на правилах і пропонує сумісність із Snort Rules , вона також забезпечує запобігання вторгненню (NIPS) та моніторинг безпеки мережі, і використовує як сигнатури, так і методи на основі аномалій для виявлення зловмисного мережевого трафіку .

На відміну від Snort, Suricata забезпечує автономний аналіз файлів PCAP за допомогою рекодера PCAP. Він також забезпечує чудовий глибоке пакування,

перевірку та узгодження шаблонів, що робить його більш ефективним для виявлення загроз та атак . Промисловість вважає Suricata сильним конкурентом для Snort, і тому їх часто порівнюють один з одним.

Suricata та snort мають ліцензію GNU GPL, та підтримують трафік IPv6, їх установка та розгортання є досить простими.

Snort та Suricata можуть працювати на усіх операційних системах (наприклад, Linux, Mac OS X, FreeBSD, OpenBSD, UNIX та Windows) і не обмежується повною мірою серверною апаратною платформою.

У таблиці 1 наведено порівняння на високому рівні між двома IDS/IPS системами та дає огляд різних параметрів, які можна зібрати. Це високорівневе порівняння показує, що три виявлення вторгнення системи мають свої переваги, і немає системи з чіткими перевага над іншими

Таблиця 3.1

Порівняння Snort та Suricata

Параметр	**Snort**	**Suricata**
Постачальник	Cisco System	OISF
Ліцензія з відкритим кодом	GNU GPL	GNU GPL
Операційні системи	Win/Unix/Mac	Win/Unix/Mac
Налаштування та використання	Легке	Середньої тяжкості
Можливості запобігання проникненню	Так	Так
Мережевий трафік	IPv4/IPv6	IPv4/IPv6
Техніка виявлення вторгнень	SIDS, AIDS	SIDS, AIDS
НаявністьGUI конфігурації	Так	Так
Підтримка високошвидкісної мережі	Середня	Висока

3.3.2 Методологія дослідження найефективнішого IDS/IPS для IoT

Як вже згадувалося раніше, забезпечити безпечне використання розумних пристроїв стає складною задачею. Це тема, в якій експерти з питань безпеки та домашньої автоматизації намагаються підтримувати баланс між використанням розумних пристроїв та високим рівнем безпеки, вимогами та апаратними обмеженнями інфраструктури. Загалом, ці середовища страждають від обмежень, які вбудовані в апаратне

забезпечення, яке обмежує їх здатність реалізувати комплексні заходи безпеки, та збільшують їх вразливість перед атаками. Щоб обрати оптимальні захисні рішення необхідно вивчити ці апаратні обмеження та переконатися, що вони не впливають на продуктивність цих рішень для захисту розумних пристроїв. Зважаючи на це ми спробуємо вивчити відомі системи виявлення вторгнень Snort та Suricata, щоб знайти найбільш підходящу для IoT з точки зору споживання ресурсів та точності виявлення. Більш конкретно, в роботі розглянуто продуктивність в режимі реального часу цих IDS/IPS при одночасному моніторингу мережевого трафіку від мережі IoT. Інформація про завантаженість процесора та оперативної пам'яті буде записана, проаналізована та порівняна.

Експерименти проводились на віртуальній машині, що працює на базі ОС CentOS 8 з 4 ГБ оперативної пам'яті, 40 ГБ жорсткого диска та на процесорі Intel Core i5-8250U, що працює на частоті 1,6 ГГц. Гіпервізор – Oracle VirtualBox версії 6.1.14, хостова ОС- Windows 10 Pro . Всі дані про про завантаження системи збирались через диспечер завдань, що влаштований в хостову ОС. У сценаріях моделювання спочатку ми робимо знімок чистої машини, перш ніж завантажувати зловмисний зразок. Потім виконуємо зловмисний зразок і фіксуємо всю інформацію, пов’язану із споживанням ресурсів та станом віртуальної машини, після цього віртуальна машина повертається до свого початкового стану. У цих експериментах кожна IDS/IPS була окремо встановлена на однакові віртуальні машини з параметрами продуктивності за замовчуванням.

Оцінка ефективності кожної IDS/IPS проводиться для 20 PCAP зразків зловмисного трафіку, генерованого різними типами атак. Файли PCAP були зібрані з ресурсу malware-traffic-analysis.net.

Одні і ті ж шкідливі файли pcap використовувались для моніторингу ресурсів, що використовуються двома різними IDS/IPS під час аналізу трафіку та генерації попереджень. TCPreplay використовується для

відтворення шкідливих файли PCAP для IDS/IPS (рис. 2). У таблиці 2 наведено Зразки PCAP зловмисного трафіку, використані в експериментах

Таблиця 3.2

Види протестованих атак

№	Тип шкідливого програмного забезпечення у файлі PCAP	Розмір файлу PCAP
1	Malspam traffic	1.0 3MB
2	Necurs Botnet Malspam	448 KB
3	Payment Slip Malspam	3.0 МБ
4	MedusaHTTP malware	669 КБ
5	Adwind Malspam	1.7 КБ
6	KainXN EK	1.93 МБ
7	Cyber Ransomware	584 КБ
8	Locky-malspam-traffic	285 КБ
9	Facebook Themed Malspam	1.4 МБ
10	BOLETO Malspam infection traffic	2.4 МБ
11	Pizzacrypt	254.4 КБ
12	BIZCN Gate Actor Nuclear	0.9 8 МБ
13	Fiesta Ek	1.52 МБ
14	Nuclear EK	2 МБ
15	Fake-Netflix-login-page-traffic	768 КБ
16	URSNIF Infection with DRIDEX	2.5 МБ
17	Dridex Spam trrafic	999 КБ
18	Brazil malware spam	12.7 МБ
19	Info stealer that uses FTP to exfiltrate data	1.4 МБ
20	Hookads-Rig-EK-sends-Dreambot	595 КБ

3.3.3 Результати дослідження

Використання оперативної пам'яті.

На рисунку 3.7 порівнюються результати використання оперативної пам'яті для кожного зразку шкідливого програмного забезпечення для двох IDS/IPS Snort, та Suricata. З отриманих результатів ми можемо зробити висновки стосовно завантаження оперативної пам’яті. Snort IDS/IPS використовує більше оперативної пам’яті для більшості зразків PCAP. Значення знаходяться в межах приблизно від 60% до 80%. Найвищі показники були записано для зразків №4, №13 та №19. Під час використання Suricata зафіксовано відносно нижчі показники, ніж у Snort, вони коливалися від 20% до 50%. Отже ми бачимо що трафік шкідливих програм а загальному має значний вплив на використання оперативної пам'яті для двох IDS/IPS, через те, що вони починають працювати більш активно, використовуючи більше ресурсів.

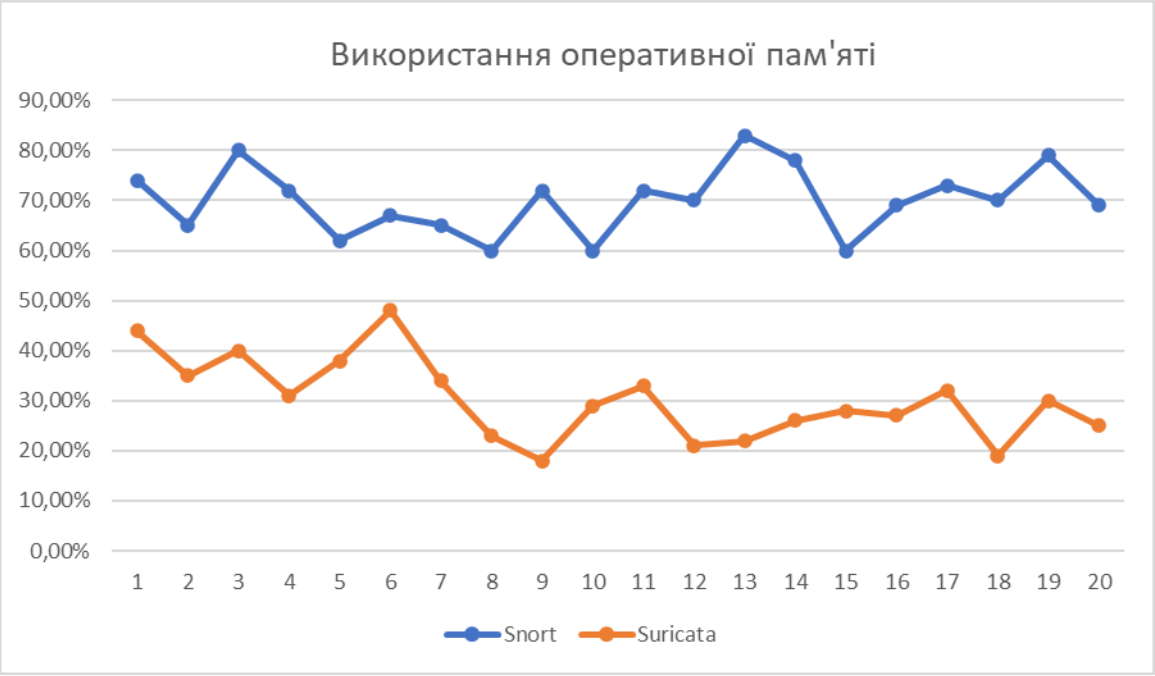


Рисунок 3.7 Графік икористання ресурсу оперативної пам’яті

Використання ресурсу процесора

На рисунку 3.8 порівнюються результати використання процесора для кожного зразка шкідливого програмного забезпечення для двох IDS/IPS Snort, Suricata та Bro IDS. З отриманих результатів спостерігається, що Snort IDS/IPS зафіксувала більші показники використання процесора для більшості PCAP від 60% до 70%. Під час використання Suricata зафіксовано відносно нижчі показники для тих самих тестів на зловмисному програмному забезпеченні. Показники використання процесора нею становить від 20% до 40. Також спостерігається, що тип трафіку шкідливого програмного забезпечення має значний вплив на використання центрального процесора, так само, як і оперативної пам'яті кожен IDS/IPS дає різні показники використання процесора для однієї і тієї ж тестової атаки, оскільки вони діють досить по-різному для кожної атаки, але все рівно при тестуванні використання ресурсів було вищим, ніж в стані спокою.

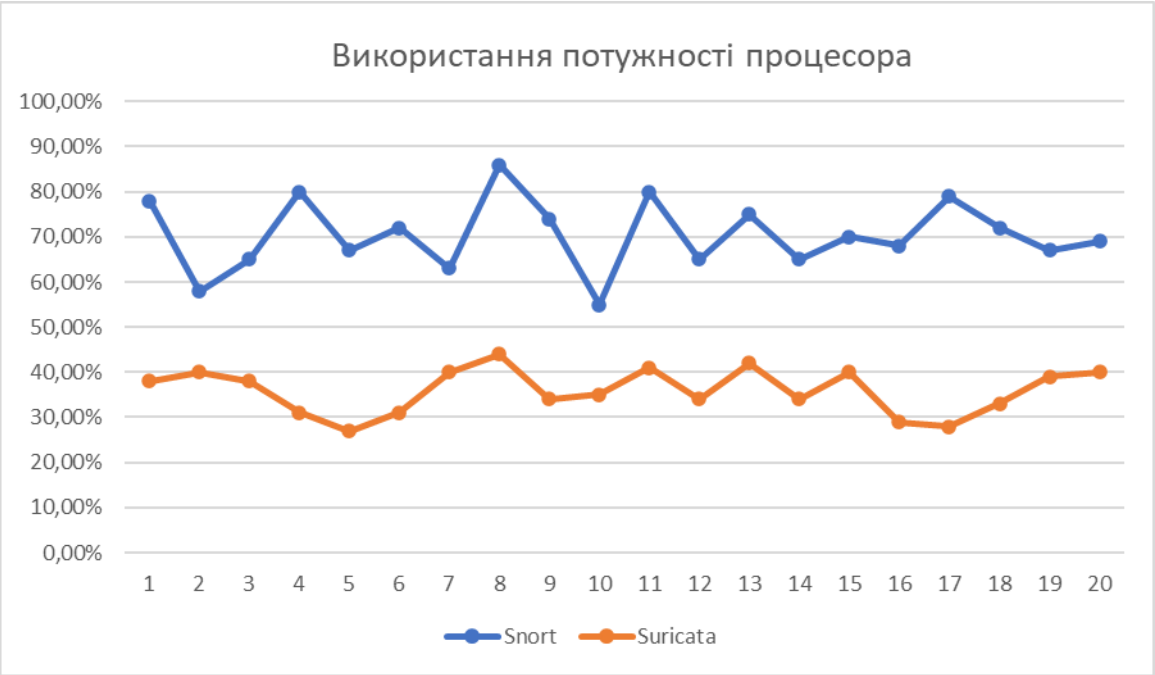


Рисунок 3.8 Графік використання ресурсу процесора

Таким чином, з цього кількісного порівняння можна зробити висновок стосовно цих IPS, з точки зору навантаження на систему та використання

ресурсів (CPU та RAM), в якому при роботі Snort використання CPU і пам'яті було відносно вищим, ніж у Suricata.. Suricata використовувала в середньому 30,5% обсягу пам’яті пам'яті, що було на 35%, меншим за використання пам'яті при робот. Snort. Отримані результати цих експериментів демонструють що Suricata працює краще, ніж Snort у випадку обмежень в апаратному забезпеченні, тому вона більше підходять для мереж розумних пристроїв.

ВИСНОВКИ

У роботі було розглянуто поняття інтернету речей, технології що він використовує. Пристрої IoT виконують різноманітні функції в різних галузях і іноді сильно відрізняються стандартами безпеки виробників. Незахищені пристрої IoT можуть стати легкою мішенню для кібератак. Було проведено аналіз основних недоліків пристроїв та мереж інтернету речей. Також було досліджено основні типи кібератак, та описано на якому рівні кожна з них може проводитись.

Було описано основні організаційні заходи, які може провести кожен користувач розумних пристроїв, а також більш детально описано технічні засоби протидії кібератакам.

У цій роботі пропонується альтернативне полегшене рішення криптографічного захисту, що поєднує характеристики симетричного та асиметричного алгоритму шифрування. Ця запропонована модель забезпечує захист потоку даних при обміні пристроями IoT.

У цій роботі було проведено порівняння продуктивності двох IPS/IDS систем з відкритим кодом Snort та Suricata, щоб знайти найбільш підходящу для IoT з точки зору споживання ресурсів, а саме центрального процесора та оперативної пам’яті. Це дослідження показало, що кожна з них має свої сильні та слабкі сторони, але згідно результатам експеременту Suricata використала в середньому на 30% менше ресурсів, ніж Snort, а саме тому вона більше підходить для мереж розумних речей, ніж Snort.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1.DDOS-АТАКА ОТ ВИДЕОНЯНИ: ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ IOT И BIG DATA [Електронний ресурс] / Анна Вичугова. – 2020. – Режим доступу до ресурсу: https://www.bigdataschool.ru/blog/iot-big-data-security-incidents.html.

3. Saeed Q. Y. Hybrid Encryption/Decryption Technique Using New Public Key and Symmetric Key Algorithm / Saeed., 2014

7. Khan, R. [and others], Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges, Frontiers of Information Technology (FIT), 2012 10th International Conference on. — 2012

8. Горященко К.Л. Архітектура ARM як потенційна основа грідінфраструктури наукової бази України

9. К.Л. Горященко, В.П. Нездоровін, Є. Г. Махрова. Вісник Хмельницького національного університету. – 2012. – №1. – С. 204-208.

10. Горященко К.Л. Риски цілісності інформації на переносних носіях інформації / К.Л. Горященко, О.І. Полікаровських, В.Є. Гавронський, Ю.І. Сніжко // Вісник Хмельницького національного університету. – 2008. – №4. – С. 66-70.

12. Albin E. A realistic experimental comparison of the Suricata and Snort intrusion-detection systems. In 2012 26th International Conference on Advanced Information Networking and Applications Workshops / E. Albin, C. Neil., 2012.

13. Williams L. A Supervised Intrusion Detection System for Smart Home IoT Devices / L. Williams, E. Anthi, M. Słowińska., 2019.

14. Brumen B. Performance analysis of two open-source intrusion detection systems. In 2016 39th International Convention on Infor-mation and Communication Technology, Electronics and Microelectronics (MIPRO) / B. Brumen, J. Legvart., 2016.

15. Bulajoul W. Network intrusion detection systems in high-speed traffic in computer networks / W. Bulajoul, A. James, P. Mandeep., 2013 , 168–175.

16. A Survey on Sensor-based Threats to Internet-of-Things (IoT) Devices and

Applications // електрон. текст. Дані :URL:https://arxiv.org/pdf/1802.02041.pdf

17. Jeong S. Internet of Things for Smart Manufacturing System: Trust Issues in Resource Allocation / S. Jeong, W. Na, J. Kim., 2018.

18. Dynamic configuration of sensors using mobile sensor hub in internet of things paradigm / C.Perera, P. Jayaraman, A. Zaslavsky, P. Christen., 2013.

19. Internet of Things, IoT, технології та стандарти Інтернету Речей [Електронний ресурс] – Режим доступу до ресурсу: https://www.it.ua/knowledge-base/technology-innovation/internet-veschej-internet-of-things-iot.