НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«На правах рукопису»
УДК 004.056

«До захисту допущено»
В.о. завідувача кафедри
_____ Микола ГРАЙВОРОНСЬКИЙ
“___”_____2020 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності: 125 Кібербезпека

на тему: Аудит даних автентифікації в Інтернеті речей _____
_____
_____

Виконав (-ла): студент (-ка) __2__ курсу, групи __ФБ-91мпв____
(шифр групи)

Скрипюк Богдан Романович _____ _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник к.т.н. Коломицев М.В. _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

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

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

Київ – 2020 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою
Спеціальність (освітньо-професійна програма) – 125 Кібербезпека («Системи, технології та математичні методи кібербезпеки)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ М.В.Грайворонський
(підпис)

«___»_____2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Скрипюку Богдану Романовичу
(прізвище, ім'я, по батькові)

1. Тема дисертації Аудит даних автентифікації в Інтернеті речей

науковий керівник дисертації к.т.н. Коломицев Михайло Володимирович ,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «15» листопада 2019 р. № 3927-с

2. Термін подання студентом дисертації 07.12.2020 р.

3. Об'єкт дослідження стан захищеності Інтернету речей

4. Вихідні дані

5. Перелік завдань, які потрібно розробити Проаналізувати архітектуру IoT, механізми безпеки та сфери застосування IoT; Ознайомитися з існуючими атаками з використанням аутентифікації, із запропонованими моделями аутентифікації та управлінням доступу в IoT, проаналізувати їх переваги і недоліки; Запропонувати метод швидкого виявлення порушень безпеки системи IoT на основі використання детектору аномалій; Реалізувати запропонований метод, підготувати результати роботи даного методу та зробити висновки.

6. Орієнтовний перелік ілюстративного матеріалу

_____

_____

7. Орієнтовний перелік публікацій IV міжнародна науков-практична конференція «Потенціал сучасної науки» 10-11 грудня 2019року _____

_____

8. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз архітектури IoT, сфери застосування	1.09-6.09	
2	Ознайомлення з існуючими схемами аутентифікації в Інтернеті речей	6.09-11.09	
3	Ознайомлення з існуючими атаками з використанням аутентифікації	10.09-17.09	
4	Ознайомлення із запропонованими моделями аутентифікації в IoT	18.09-25.09	
5	Аналіз питання безпеки IoT	26.09-3.10	
6	Розробка методу швидкого виявлення порушень безпеки системи IoT на основі використання детектору аномалій	4.10-25.10	
7	Реалізація методу, підготовка результатів роботи методу	26.10-14.11	

Студент _____ _____
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ _____
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Представлена робота обсягом __ сторінки містить 3 ілюстрацій, 22 таблиці та 18 літературних посилань.

Метою даної роботи є аналіз стану захищеності систем Інтернету речей, розробка методу виявлення атак шляхом аналізу даних та виявлення в них підозрілої активності.

Об’єктом дослідження є стан захищеності Інтернету речей.

Предметом дослідження є виявлення та виправлення проблем безпеки архітектури Інтернету речей, схеми автентифікації присутні в IoT.

Методами дослідження було обрано: ознайомлення та опрацювання літературних джерел, що представлено монографічними та журнальними матеріалами, електронними ресурсами, котрі стосуються даної теми, аналіз архітектури та стану захищеності систем Інтернету речей, аналіз атак на автентифікацію в Інтернеті речей для подальшого використання при розробці методу.

Наукова новизна полягає в тому, що розроблений метод є унікальним та не має аналогів, він дозволяє швидко та ефективно виявити загрози безпеки системи Інтернету речей з боку автентифікації. Даний метод використовує детектор аномалій для виявлення загроз, що ніким не описано і не реалізовано.

Практичне значення результатів роботи випливає з можливості використання даного методу для побудови системи виявлення атак, яка базується на процесі автентифікації, а також на основі використання детектору аномалій для виявлення відхилень в даних автентифікації.

Ключові слова: Інтернет речей, атака, детектор аномалій, автентифікація, датасет.

ABSTRACT

The work includes __ pages, 3 figures and 18 literary references and 22 tables.

The aim of this qualification work is to analyze the security of IoT systems, to develop a method of detecting attacks by analyzing data and identifying suspicious activity in them.

The object of research is the security of the Internet of Things.

The subject of research is the problems of the IoT security architecture and authentication schemes present in IoT.

Methods of research: acquaintance and processing of the literary sources presented by monographic and journal materials, the electronic resources concerning the given theme, the analysis of architecture and a condition of safety of systems of the Internet of things, the analysis of attacks on authentication in the Internet of things for the further use at the development of a method.

The scientific novelty is that the developed method is unique and has no analogues, it allows you to quickly and effectively identify threats to the security of the IoT system from the side of authentication. This method uses an anomaly detector to detect threats, is not described or implemented by anyone.

The practical value of the work follows from the possibility of using this method to build an attack detection system, which is based on the authentication process, as well as on the use of an anomaly detector to detect deviations in the authentication data.

Keywords: Internet of Things, attack, anomaly detector, authentication, dataset.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ВСТУП

У 2017 році кількість пристроїв, підключених до інтернету, за даними Statista, перевищила 20 млрд штук. До 2020 їх буде вже близько 50 млрд, передбачає Cisco. Інтернет речей - це не тільки виконавчий холодильник, який сам замовляє улюблену їжу господаря, або послужливий чайник, який кип'ятить воду на першу вимогу зі смартфона. Це розумні датчики на полях, дрони з камерами, завдяки яким можна віддалено моніторити стан грунтів, це датчики в громадському транспорті і єдині системи для моніторингу життя міста. Іншими словами, вже через кілька років інтернетом речей стане світ навколо нас.

Використання систем IoT в повсякденному житті людей може зробити питання безпеки життя дуже актуальним. Інтегрована в будинку, автомобілі та електромережі інтелектуальність може бути використана хакерами для створення небезпечних ситуацій. Різні сценарії злому, представлені в останні роки, ілюструють рівень шкоди, яку може бути заподіяно порушеннями безпеки, особливо при розробці і широкому впровадженні застосунків IoT, що працюють з конфіденційною інформацією.

Актуальність роботи. Актуальність роботи зумовлена тим, що ринок IoT пристроїв не так давно почав розвиватися і приносити нові загрози в інформаційні мережі. Найближчим часом модулі Інтернету речей будуть розміщені в кожному домі, а наслідки, створені атаками, можуть бути непередбачуваними.

Об’єктом дослідження є стан захищеності Інтернету речей.

Предметом дослідження є виявлення та виправлення проблем безпеки архітектури Інтернету речей, схеми автентифікації присутні в IoT.

Метою роботи є аналіз стану захищеності систем Інтернету речей, розробка методу виявлення атак шляхом аналізу даних та виявлення в них підозрілої активності.

Завдання роботи:

1. Проаналізувати архітектуру IoT, механізми безпеки та сфери застосування IoT.
2. Ознайомитися з існуючими схемами автентифікації в Інтернеті речей, з існуючими атаками з використанням автентифікації, із запропонованими моделями автентифікації та управлінням доступу в IoT, проаналізувати їх переваги і недоліки.
3. Запропонувати метод швидкого виявлення порушень безпеки системи IoT на основі використання детектору аномалій.
4. Реалізувати запропонований метод, підготувати результати роботи даного методу та зробити висновки.

Методи дослідження - ознайомлення та опрацювання літературних джерел, що представлено монографічними та журнальними матеріалами, електронними ресурсами, котрі стосуються даної теми, аналіз архітектури та стану захищеності систем Інтернету речей, аналіз атак на автентифікацію в Інтернеті речей для подальшого використання при розробці методу.

Наукова новизна полягає в тому, що розроблений метод є унікальним та не має аналогів, він дозволяє швидко та ефективно виявити загрози безпеки системи Інтернету речей з боку автентифікації. Даний метод використовує детектор аномалій для виявлення загроз, що ніким не описано і не реалізовано.

Результати роботи викладені у третьому розділі, вони демонструють практичну роботу нового методу, який буде швидко аналізувати дані та виявляти пристрої через які відбуваються спроби отримання несанкціонованого доступу до системи.

Практичне значення результатів роботи випливає з можливості використання даного методу для побудови системи виявлення атак, яка базується на процесі автентифікації, а також на основі використання детектору аномалій для виявлення відхилень в даних автентифікації.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ІНТЕРНЕТУ РЕЧЕЙ

1.1 Роль Інтернету речей в сучасних IT технологіях

У зв'язку з бурхливим розвитком мереж з пакетною комутацією і перш за все Інтернету на початку 2000-х років світове телекомунікаційне співтовариство спочатку виробило, а потім і приступило до реалізації нової парадигми розвитку комунікацій - мереж наступного покоління NGN (Next Generation Networks). Технології NGN вже пройшли еволюційний шлях розвитку від гнучких комутаторів (Softswitch) до підсистем мультимедійної зв'язку IMS (IP Multimedia Subsystem) і бездротових мереж тривалої еволюції LTE (Long Term Evolution). При цьому завжди передбачалося, що основними користувачами мереж NGN будуть люди і, отже, максимальне число абонентів в таких мережах завжди буде обмежена чисельністю населення планети Земля. [1]

Однак останнім часом значного розвитку набули методи радіочастотної ідентифікації RFID (Radio Frequency IDentification), бездротові сенсорні мережі WSN (Wireless Sensor Network), комунікації малого радіусу дії NFC (Near Field Communication) і міжмашинні комунікації M2M (Machine-to-Machine), які, інтегруючись з інтернетом, дозволяють забезпечити простий зв'язок різних технічних пристроїв («речей»), число яких може бути величезним. За розрахунками консалтингового підрозділу Cisco IBSG в проміжку між 2008 і 2009 роками кількість підключених до інтернету предметів перевищило кількість людей, до 2015 року кількість підключених пристроїв досягне 25 мільярдів, а до 2020 року - 50 мільярдів. Таким чином, в даний час відбувається еволюційний перехід від «Інтернету людей» до «Інтернету речей», IoT (Internet of Things). [1]

У загальному випадку під Інтернетом речей розуміється сукупність різноманітних приладів, датчиків, пристроїв, об’єднаних в мережу за

допомогою будь-яких доступних каналів зв'язку, що використовують різні протоколи взаємодії між собою і єдиний протокол доступу до глобальної мережі. У ролі глобальної мережі для Інтернет-речей зараз використовується мережа Інтернет. Спільним протоколом є IP. [1]

Слід особливо відзначити, що Інтернет речей не виключає участь людини. IoT в повному обсязі автоматизує речі, так як він орієнтований на людину і надає йому можливість доступу до речей. Але багато речей зможуть вести себе інакше, ніж ми уявляємо собі сьогодні. У IoT кожна річ має свій унікальний ідентифікатор, які спільно утворюють континуум речей, здатних взаємодіяти один з одним, створюючи тимчасові або постійні мережі. Так речі можуть брати участь в процесі їх переміщення, ділячись інформацією про поточну геопозіції, що дозволяє повністю автоматизувати процес логістики, а маючи вбудований інтелект, речі можуть змінювати свої властивості і адаптуватися до навколишнього середовища, в тому числі для зменшення енергоспоживання. Вони можуть виявляти інші, так чи інакше пов'язані з ними речі, і налагоджувати з ними взаємодію. IoT дозволяє створювати комбінацію з інтелектуальних пристроїв, об'єднаних мережами зв'язку, і людей. Спільно вони можуть створювати найрізноманітніші системи, наприклад, для роботи в середовищах, незручних або недоступних для людини (в космосі, на великій глибині, на ядерних установках, в трубопроводах і т.п.). [1]

Вважається, що першу в світі інтернет-річ створив один з батьків протоколу TCP/IP Джон Ромкі в 1990 році, коли він підключив до мережі свій тостер. Але тільки в 21 столітті в зв'язку з бурхливим розвитком інформаційно-комунікаційних технологій сформувалася концепція IoT і отримала своє практичне втілення. Все почалося з необхідності оптимізації системи логістики та управління системою постачання підприємств. Друга хвиля інновацій була обумовлена необхідністю скорочення витрат в системах спостереження, безпеки, транспорту та ін. Третя була викликана потребою в геолокаційних сервісах. Четверта хвиля буде обумовлена необхідністю дистанційної

присутності людини на місці події і вимагатиме його уваги, яка стане можливим завдяки мініатюрним вбудованим процесорам. [1]

Інтернет речей розвивається і більша кількість предметів буде підключатися до глобальної мережі, створюючи таким чином більше можливостей в сфері безпеки, управління і аналітики, створюючи нові і ширші перспективи і сприяючи підвищенню рівня життя населення. Прогнозується, що «речі» в подальшому будуть активними учасниками бізнесу, соціальних та інформаційних процесів, де вони матимуть можливість спілкуватися та взаємодіяти один з одним, обмінюватись даними про навколишнє середовище, реагувати і впливати на процеси, котрі відбуваються навколо, без втручань з боку людини.

1.2 Базові принципи IoT

Інтернет речей має три основних принципи. По-перше, повсюдно поширену комунікаційну інфраструктуру, по-друге, глобальну ідентифікацію кожного об’єкта і, по-третє, можливість кожного об’єкта відправляти та отримувати дані за допомогою персональної мережі або мережі Інтернет, до якої він підключений. [2]

Найбільш важливими відмінностями Інтернету речей від існуючого інтернету людей є:

- фокус на речах, а не на людині;
- істотно більша кількість підключених об’єктів;
- істотно менші розміри об’єктів і невисокі швидкості передачі даних;
- фокус на зчитуванні інформації, а не на комунікаціях;
- необхідність створення нової інфраструктури і альтернативних стандартів. [2]

1.3 Архітектура IoT

Архітектура IoT включає чотири функціональних рівня (рисунок 1.1), описаних нижче.

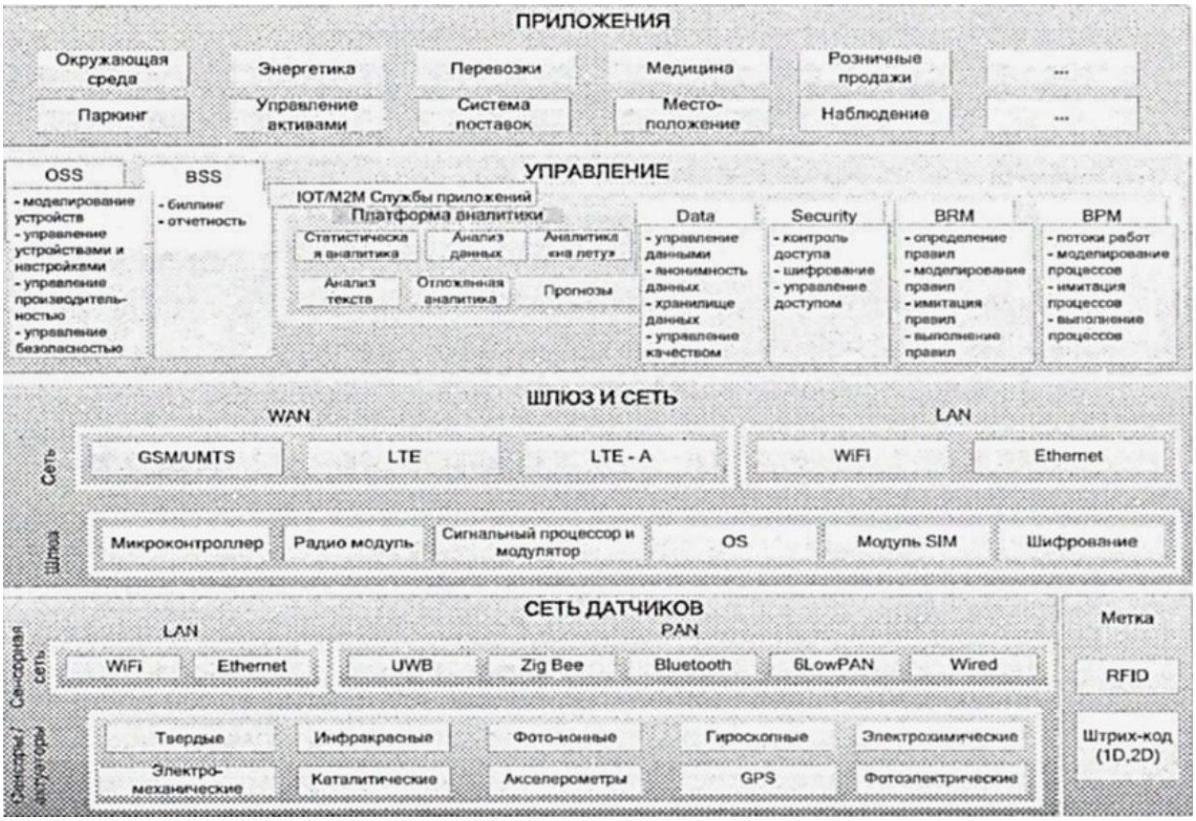

Рисунок 1.1 - Архітектура Інтернету речей

1.3.1 Рівень сенсорів і сенсорних мереж

Цей рівень архітектури є найнижчим та складається з «розумних» об’єктів, інтегрованих з датчиками. Сенсори використовуються для реалізації з’єднання цифрового та фізичного світів, вони забезпечують збір та обробку даних в реальному часі. Зменшення розмірів апаратних сенсорів дозволило інтегрувати їх зокрема в об’єкти фізичного світу. Існує багато типів сенсорів для різних цілей, наприклад сенсои що вимірюють тиск, температуру, швидкість переміщення, місце розташування. Сенсори можуть мати невелику

пам'ять, даючи можливість записувати кілька результатів вимірювань. Сенсор може вимірювати фізичні параметри контрольованого об'єкта/явища і перетворювати їх в сигнал, який може бути прийнятий відповідним пристроєм. Сенсори класифікуються відповідно до їх призначення, наприклад, сенсори навколишнього середовища, сенсори для тіла, сенсори для побутової техніки, сенсори для транспортних засобів і т. п. [1]

Більшість сенсорів вимагає з'єднання з агрегатором сенсорів (шлюзом), які можуть бути реалізовані з використанням локальної обчислювальної мережі (LAN, Local Area Network), таких як Ethernet і Wi-Fi або персональної мережі (PAN, Personal Area Network), таких як ZigBee , Bluetooth і ультраширокополосного бездротового зв'язку на малих відстанях (UWB, Ultra-Wide Band). Для сенсорів, які не вимагають підключення до агрегатора, зв'язок з серверами/додатками може надаватися з використанням глобальних бездротових мереж WAN, таких як GSM, GPRS і LTE[3].

Сенсори, які характеризуються низьким енергоспоживанням і низькою швидкістю передачі даних, утворюють широко відомі бездротові сенсорні мережі (WSN, Wireless Sensor Network). [1]

1.3.2 Рівень шлюзів і мереж

Великий обсяг даних, що створюється на першому рівні цифрового двійника численними мініатюрними сенсорами, вимагає надійної та високопродуктивної проводової або бездротової мережевої інфраструктури в якості транспортного середовища. Існуючі мережі зв'язку, що використовують різні протоколи, можуть бути використані для підтримки міжмашинних комунікацій M2M і їх додатків. Для реалізації широкого спектру послуг і додатків в IoT необхідно забезпечити спільну роботу безлічі різних технологій і протоколів доступу в гетерогенній конфігурації. Ці мережі повинні забезпечувати необхідні значення якості передачі інформації, і перш за все по

затримці, пропускній спроможності і безпеці. Даний рівень складається з конвергентної мережевої інфраструктури, яка створюється шляхом інтеграції різнорідних мереж в єдину мережеву платформу. Конвергентний абстрактний мережевий рівень дозволяє через відповідні шлюзи декільком користувачам використовувати ресурси в одній мережі незалежно і спільно без шкоди для конфіденційності, безпеки і продуктивності[1].

1.3.3 Сервісний рівень

Сервісний рівень містить набір інформаційних послуг, покликаних автоматизувати технологічні і бізнес операції двійників: підтримки операційної і бізнес діяльності (OSS / BSS, Operation Support System / Business Support System), різної аналітичної обробки інформації (статистичної, інтелектуального аналізу даних і текстів, прогностичної аналітики та ін.), зберігання даних, забезпечення інформаційної безпеки, управління бізнес-правилами (BRM, Business Rule Management), управління бізнес-процесами (BPM, Business Process Management) та ін. [1]

1.3.4 Рівень застосунків

На четвертому рівні архітектури існують різні типи додатків для відповідних промислових секторів і сфер діяльності (енергетика, транспорт, торгівля, медицина, освіта та ін.). Додатки можуть бути «вертикальними», коли вони є специфічними для конкретної галузі промисловості, а також «горизонтальними», (наприклад, управління автопарком, відстеження активів та ін.), Які можуть використовуватися в різних секторах економіки. [1]

1.4 Взаємодія IoT з перспективними інфокомунікаційними технологіями

Важливу роль у становленні та успішному впровадженні Інтернету речей відіграють різні перспективні інфокомунікаційні технології, такі як великі дані, хмарні технології і повсюдна комп'ютеризація, з якими IoT активно взаємодіє.

1.4.1 Великі дані(Big Data)

До початку XX століття обсяг знань подвоювався кожне сторіччя, сьогодні обсяг знань людства подвоюється кожні 2-3 роки. 70% всієї доступної інформації з'явилося після винаходу Інтернету. Інтернет речей радикальним чином збільшує обсяг зібраних даних, що є наслідком величезної кількості джерел інформації (насамперед різні сенсори). Гігантські сенсорні мережі виробляють величезні потоки даних, які треба вміти не тільки зберігати, але й обробляти, робити по ним висновки, приймати рішення - і все це з урахуванням неточності як оригінальних даних, так і процедур обробки. В кінці 2000-х років для обробки великого обсягу даних сформувався підхід, який називається «великі дані» (англ. Big Data) - це серія інструментів і методів обробки структурованих і неструктурованих даних величезних об’ємів і значного різноманіття для отримання необхідних результатів обробки. В якості визначальних характеристик для великих даних відзначають «три V»: об’єм (англ. Volume, в сенсі величини фізичного об’єма), швидкість (англ. Velocity, в сенсах як швидкості приросту, так і необхідності високошвидкісної обробки і отримання результатів), різноманіття (англ. variety, в сенсі можливості одночасної обробки різних типів структурованих і неструктурованих даних).(рисунок 1.2)

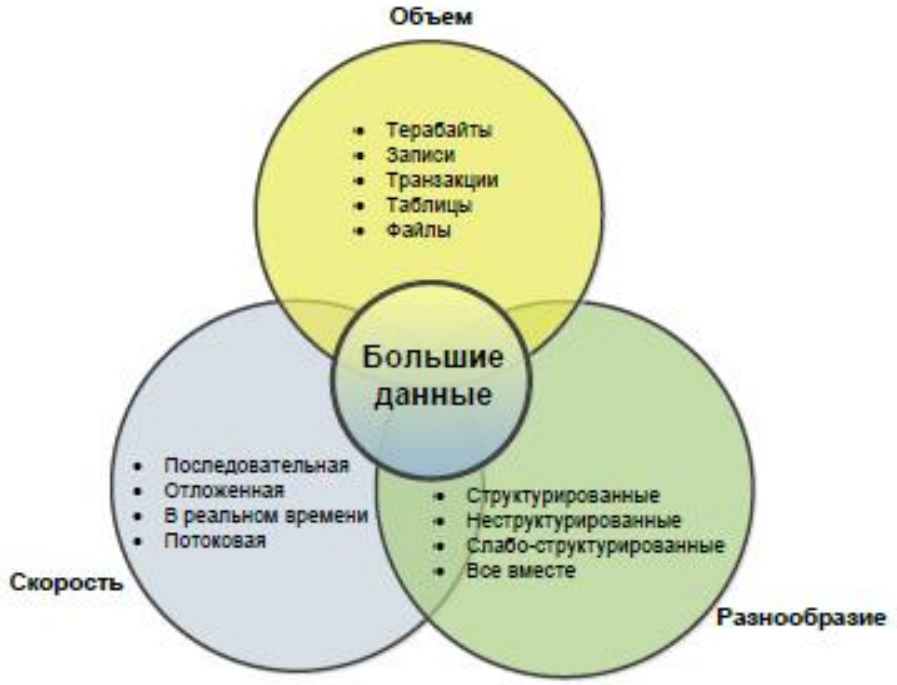

Рисунок 1.2 - три основні характеристики великих даних

Основна відмінність великих даних від «звичайних» полягає в тому, що ці дані неможливо обробити традиційними системами управління базами даних (СКБД) і рішеннями класу Business Intelligence через їх великий обсягу і різноманітний склад. Інша важлива їх властивість - феноменальне прискорення накопичення даних і постійна зміна. Такі популярні завдання, як зведення даних, отриманих з різних джерел (Data Cleaning, Data Merging, De-deduplication), вимагають особливих методів аналізу в разі неточних даних, особливо даних величезних розмірів. У зв'язку з цим і був розроблений набір інструментів, який отримав назву «великі дані», що дозволяють працювати з даними незалежно від їх типу і обсягу.

Прогнозується, що запровадження технологій великих даних матиме найбільший вплив на інформаційні технології у виробництві, охороні здоров'я, торгівлі, державному управлінні, а також в сферах і галузях, де реєструються індивідуальні переміщення ресурсів і де потенційно можуть бути використані технології Інтернету речей. [7]

1.4.2 Хмарні обчислення

Оскільки Інтернет речей породжує «великі дані», то виникає закономірне питання: де їх зберігати і чим обробляти? Відповіддю на це питання є перспективна інфокомунікаційна технологія - хмарні обчислення (CC, Cloud Computing). Хмарні обчислення мають на увазі оренду послуг і ресурсів для зберігання і обробки даних в глобальній мережі замість власної інфраструктури. У систем CC повинні бути п'ять основних характеристик: самообслуговування на вимогу, широкосмуговий мережевий доступ, пул ресурсів, можливість швидкого перенастроювання або розширення і вимірюване обслуговування.

Існують чотири моделі развертання хмарної інфраструктури (так званих «хмар»): Приватна хмара, Публічна хмара, Гібридна хмара, Суспільна хмара.

Різні послуги CC, що позначаються в загальному випадку як XaaS (X as a Service), можна віднести до трьох основних класів:

- «Інфраструктура, як послуга» (IaaS, Infrastructure as a Service) - оренда потужності серверів і ємності систем зберігання центрів обробки даних (ЦОД);
- «Програмне забезпечення, як послуга» (SaaS, Software as a Service) – оренда програмного забезпечення (ПЗ), яке запускається «з хмари»;
- «Платформа, як послуга» (PaaS, Platform as a Service) - оренда платформи розробки ПЗ колективними або індивідуальними розробниками.

Для роботи технологій Інтернету речей можна використовувати і туманні обчислення (Fog Computing). Під «туманом» мається на увазі наближення «хмари» до землі, в даному випадку «туман» - це різновид хмарних сервісів, розташованих не десь в недоступних висотах, а в навколишньому середовищі. Інакше кажучи, Fog Computing не альтернатива, а доповнення до Cloud

Computing, і можуть виникнути ситуації їх спільної дії (наприклад, виконання аналітичного застосунку), і в такому випадку Cloud надасть послугу Fog.

Туманні обчислення доповнюють хмарні обчислення і забезпечують взаємодію розумних речей між собою і хмарними ЦОД у вигляді трирівневої ієрархічної структури. Верхній рівень займають тисячі хмарних ЦОД, що надають ресурси, необхідні для виконання серйозних, наприклад аналітичних, програмних застосунків IoT. Рівнем нижче розташовуються десятки тисяч розподілених керуючих ЦОД, в яких міститься «інтелект» Fog Computing, а на нижньому рівні знаходяться мільйони обчислювальних пристроїв розумних речей.

Fog Computing можна визначити як в максимальному ступені віртуалізовану платформу, що підтримує три основних типи сервісів, що утворюють міжмашинні комунікації M2M: обчислення, зберігання та мережу. Завдання Fog Computing полягає в забезпечення взаємодії мільярдів пристроїв між собою і з хмарними ЦОД.

1.5 Напрямки практичного застосування IoT

На основі Інтернету речей можуть бути реалізовані всілякі «розумні» (smart) додатки в різних сферах діяльності і життя людини:

«Розумна планета» - людина зможе буквально «тримати руку на пульсі» планети: своєчасно реагувати на упущення в плануванні господарств, забруднення та інші екологічні проблеми, а значить, ефективно розпоряджатися невідновлюваними ресурсами.

«Розумне місто» - міська інфраструктура і супутні муніципальні послуги, такі як освіта, охорона здоров'я, громадська безпека, ЖКГ, стануть більше пов'язаними і ефективними.

«Розумний будинок» - система буде розпізнавати конкретні ситуації, що відбуваються в будинку, і реагувати на них відповідним чином, що забезпечить мешканцям безпеку, комфорт і ресурсозбереження.

«Розумна енергетика» - буде забезпечена надійна і якісна передача електричної енергії від джерела до приймача в потрібний час і в необхідній кількості.

«Розумний транспорт» - переміщення пасажирів з однієї точки простору в іншу стане зручнішим, швидшим і безпечнішим.

«Розумна медицина» - лікарі і пацієнти зможуть отримати віддалений доступ до дорогому медичного обладнання або до електронної історії хвороби в будь-якому місці, буде реалізована система віддаленого моніторингу здоров'я, автоматизована видача лікарських препаратів хворим і багато іншого. [8]

1.6 Хмарні обчислення та IoT

Основна проблема для інтернету речей, якщо говорити не про домашнє застосування, а про бізнес або державу, це ресурси для зберігання і обробки інформації. Власна інфраструктура вимагає великих інвестицій відразу ж, причому будувати її доведеться «про запас», частина обладнання буде просто простоювати. Крім того, на будівництво потрібен час, і мова йде про місяці - на тендери, закупівлю, встановлення та налаштування обладнання. Також потрібна команда - а це і час на пошук, і витрати на зарплату.

Тому в еру інтернету речей багато компаній звернули свою увагу на хмарні технології. Вони вимагають мінімум часу на розгортання, дозволяють купувати рівно стільки ресурсів, скільки необхідно компанії на даний момент. Хмари - це практично необмежений ресурс для збору і аналізу великих даних. Аналіз дозволяє знаходити приховані закономірності і є джерелом інсайтів, які допомагають бізнесу (або державі, якщо мова йде, наприклад, про дані міст)

розвиватися, оптимізувати витрати, знаходити потенційні загрози і слабкі місця, вирішувати проблеми несподіваним чином.

Якщо говорити про хмарні рішення, то їх надають як глобальні гравці такі як Microsoft, Google, Amazon, так і регіональні, які працюють в декількох країнах. У кожного є свої переваги і недоліки. Якщо говорити про перші, то в числі переваг - відомий і популярний бренд, відсутність ризиків, пов'язаних з українською специфікою, вже досить велика кількість фахівців, які мали досвід роботи з такими хмарами. Серед недоліків - менша гнучкість у підтримці. Крім того, власник даних часто не має уявлення, де саме вони зберігаються - в якій юрисдикції і хто має до них доступ. Якщо говорити про регіональних гравців, то їх сильні сторони - це якраз підтримка з розумінням специфіки регіону, індивідуальний підхід і краще заглиблення в бізнес клієнта.

1.7 Протоколи IoT

Говорячи про Інтернет речей, ми завжди думаємо про комунікацію. Взаємодія між датчиками, пристроями, шлюзами, серверами і застосунками користувачів є важливою характеристикою, яка робить Інтернет речей тим, чим він є. Але що дозволяє всім цим інтелектуальним пристроям спілкуватися і взаємодіяти, так це протоколи IoT, які можна розглядати як мови, що використовуються устаткуванням IoT для спілкування.

Щоб висловити вищенаведену концепцію на прикладі, відмінною рисою інтелектуального пристрою від звичайного є те, що в той час як останній залишається німим в разі аварії, перший може говорити з іншими пристроями (і не тільки з пристроями того ж типу), якщо у нього виникають якісь проблеми і, при необхідності, повідомити про несправності користувачеві або автоматично звернутися за допомогою. Але кожен такий випадок взаємодії можливий тільки при наявності засобу комунікації, загальної "мови", якою всі пристрої в даній екосистемі Інтернету речей могли б користуватися. В рамках інтернету речей

середина забезпечується протоколами IoT: або вже давно використовуються протоколи інтернету, або протоколи IoT, спеціально розроблені для зв'язку з підключеними пристроями.

Протоколи IoT є важливою частиною стека технологій IoT, без них апаратне забезпечення стає марним, оскільки протоколи IoT дозволяють здійснювати структурований і значимий обмін даними. З переданих фрагментів даних можна витягти корисну інформацію для кінцевого користувача, завдяки чому все розгортання стає економічно вигідним.

Це одна з причин, по якій Інтернет речей потребує стандартизованих протоколів IoT. Вони допомагають уникнути подальшої роздробленості, зводячи тим самим до мінімуму ризик виникнення загроз безпеці.

Хоча, схоже, всі згодні з твердженням, що до сих пір було зроблено мало зусиль, щоб запропонувати світовий стандарт, який би уніфікував всю комунікацію в рамках Інтернету речей. Проте, за останні кілька років в Інтернеті речей з'явилися протоколи, які направлені на те, щоб відповісти на виклик і запропонувати універсальність без компромісів щодо безпеки та швидкості і простоти розгортання. Одним з таких протоколів IoT для задоволення конкретних потреб в різних випадках використання управління пристроями для надання відповідних призначенню рішень, одночасно пропонуючи універсальний стандарт, є протокол OMA Lightweight M2M.

З іншого боку, фрагментація Інтернету речей є результатом самої природи Інтернету речей: неоднорідність Інтернету речей, представлена безліччю його технологій і стандартів, відповідає різноманітності речей у світі, до яких прагне Інтернет речей. Аналогічним чином, існує безліч аспектів взаємодії в IoT, кожен з яких має свій тип протоколів відповідно до своїх цілей. Протоколи IoT можна розділити по тій ролі, яку вони грають в мережі. Серед багатьох інших, є протоколи, які використовуються в інфраструктурі зв'язку (наприклад, 6LowPAN), комунікації (Wi-Fi, Bluetooth), передачі даних (MQTTT, CoAP, XMPP), безпеки (DTLS) і управління пристроями, а також телеметрії (LwM2M) .

1.7.1 Протокол обмеженого застосунку (CoAP)

Хоча існуюча інфраструктура Інтернету вільно доступна і може бути використана для будь-якого пристрою Інтернету речей, вона часто виявляється занадто важкою і енергоємкою для більшості випадків використання Інтернету речей. Створений робочою групою IETF Constrained RESTful Environments і запущений в 2013 році, протокол Constrained Application Protocol (CoAP) був розроблений для перекладу моделі HTTP таким чином, щоб її можна було використовувати в обмежувальних пристроях і мережевих середовищах.

Розроблений для задоволення потреб систем Інтернету речей на базі HTTP, CoAP покладається на протокол User Datagram Protocol (UDP) для створення безпечного зв'язку між кінцевими точками. Дозволяючи мовлення і багатоадресну передачу, UDP може передавати дані на кілька хостів, зберігаючи при цьому швидкість зв'язку та низьку пропускну здатність, що робить його придатним для бездротових мереж, зазвичай використовуваних в умовах обмежених ресурсів M2M. Ще одна річ, яку CoAP розділяє з HTTP, це RESTful архітектура, яка підтримує модель взаємодії запит / відповідь між кінцевими точками програми. Більш того, CoAP приймає основні HTTP методи отримання, відправлення, встановлення і видалення, завдяки яким можна уникнути двозначності під час взаємодії між клієнтами.

Як і MQTT, CoAP має функцію Quality of Service, яка використовується для управління відправленими повідомленнями і відзначає їх як "підтверджувані" або "непідтверджувані" відповідно, що вказує на те, чи повинен одержувач повертати "ack" чи ні. Іншою цікавою особливістю CoAP є те, що він підтримує переговори по контенту і механізм виявлення ресурсів. Крім передачі даних IoT, CoAP підтримує безпеку на транспортному рівні (DTLS) для безпечного обміну повідомленнями на транспортному рівні. CoAP повністю задовольняє потреби надзвичайно легких протоколів, щоб задовольнити потреби пристроїв, що працюють від батарей або з низьким

енергоспоживанням. В цілому, CoAP добре підходить, коли мова йде про існуючі системи Інтернету речей на базі вебслужб.

1.7.2 Передача телеметрії черги повідомлень (MQTT)

Ймовірно, на сьогоднішній день найбільш поширеним стандартом в Промисловому Інтернеті речей є телеметричний протокол Message Queuing Telemetry Transport, що представляє собою легкий протокол обміну повідомленнями типу публікації / підписки (pub / sub). Архітектура MQTT, розроблена для пристроїв з живленням від батарей, проста і легка, забезпечуючи низьке енергоспоживання пристроїв. Працюючи над протоколом TCP / IP, він був спеціально розроблений для ненадійних мереж зв'язку з метою вирішення проблеми зростаючого числа невеликих дешевих малопотужних об'єктів, що з'явилися в мережі в останні роки.

MQTT основана на моделі абонента, видавця і брокера. В рамках моделі задача видавця полягає в зборі даних і відправленні інформації передплатникам через посередницький рівень, яким є брокер. З іншого боку, роль брокера полягає в забезпеченні безпеки шляхом перехресної перевірки повноважень видавців і передплатників. MQTT пропонує три способи досягнення цього (Quality of Service), завдяки яким видавець має можливість визначати якість свого повідомлення:

- QoS0 (не більше одного разу): Найменш надійний режим, але і найшвидший. Публікація відправлена, але підтвердження не отримано.
- QoS1 (не рідше одного разу): Забезпечує доставку повідомлення хоча б один раз, але може бути отриманий дублікат.
- QoS2 (рівно один раз): Найнадійніший режим при найбільшому споживанні смуги пропускання. Дублювання контролюється, щоб переконатися, що повідомлення доставлено тільки один раз.

Виявивши широке застосування в таких приладах IoT, як електролічильники, транспортні засоби, детектори, промислове і санітарне обладнання, MQTTT добре реагує на такі потреби:

- Мінімальна смуга пропускання
- Робота через бездротові мережі
- низький рівень споживання енергії
- Гарна надійність при необхідності
- Маленькі ресурси обробки та пам'яті

Незважаючи на свої особливості, MQTT може бути проблематичним для деяких дуже обмежувальних пристроїв через факт передачі повідомлень по TCP і управління довгими назвами тем. Це вирішується за допомогою варіанту MQTTT-SN, який використовує UDP і підтримує індексування імен тем. Однак, незважаючи на широке впровадження, MQTT не підтримує чітко визначену модель представлення даних і структури управління пристроями, що робить реалізацію можливостей управління даними і пристроями цілковито залежними від платформи або виробника.

1.7.3 Протокол розширеного обміну повідомленнями та присутності (XMPP)

Розроблений в 1999 році спільнотою Jabber з відкритим вихідним кодом і спочатку призначений для обміну повідомленнями в режимі реального часу, цей комунікаційний протокол IoT для проміжного програмного забезпечення, орієнтований на повідомлення, заснований на мові XML. Він дозволяє в режимі реального часу обмінюватися структурованими, але розширюваними даними між двома або більше мережевими клієнтами.

З моменту свого створення XMPP широко застосовується в якості протоколу зв'язку. Згодом, з появою легкої специфікації XMPP: XMPP-IoT, він продовжував використовуватися в контексті Інтернету речей. Будучи відкритим

стандартом, підтримуваним спільнотою, XMPP IoT має переваги адресації і масштабованості, що робить його ідеальним для розгортання IoT, орієнтованих на споживача.

Серед недоліків використання XMPP в IoT комунікації слід відзначити те, що він не забезпечує ні якості обслуговування, ні наскрізного шифрування. У зв'язку з цими обмеженнями, серед іншого, прогнозується, що його застосування в IoT залишиться слабо пов'язаним з галуззю, оскільки протокол безумовно не стане стандартом, використовуваним для обміну даними та управління пристроями з обмеженими ресурсами, як це роблять MQTT або LwM2M .

1.7.4 Служба розподілу даних (DDS)

Як і XMPP, протокол DDS був розроблений на основі методології публікації-підписки. Протокол DDS, розроблений Object Management Group (OMG), забезпечує масштабований, надійний, високопродуктивний і функціонально сумісний обмін даними між підключеними пристроями незалежно від апаратного забезпечення і програмної платформи в режимі реального часу. На відміну від протоколів MQTTT і CoAP IoT, DDS підтримує безброкерську архітектуру і мультикастинг для забезпечення високої якості QoS і функціональної сумісності.

Архітектура протоколу DDS заснована на рівні публікації-підписки (DCPS) і додатковому рівні локальної реконструкції даних (DLRL), орієнтованому на дані. У той час як рівень DCPS відповідає за ефективний і масштабований розподіл даних між абонентами з урахуванням наявних ресурсів, DLRL пропонує інтерфейс для функцій DCPS, що дозволяє передавати дані між підключеними до IoT об'єктами.

Хоча DDS не є типовим рішенням IoT, він все ще знаходить застосування в деяких галузях промисловості, таких як управління повітряним рухом, інтелектуальне управління мережами, автономні транспортні засоби, транспортні системи, робототехніка, виробництво електроенергії і медичні послуги. В цілому DDS може використовуватися для управління обміном даними між легкими пристроями і з'єднанням великих високопродуктивних сенсорних мереж. Він також може відправляти і приймати дані з хмари.

1.7.5 Розширений протокол черги повідомлень (Advanced Message Queuing Protocol, AMQP)

AMQP - це відкритий стандартний протокол типу публікація / підписка, розроблений в 2003 році, який бере свій початок в секторі фінансових послуг. Незважаючи на те, що він завоював популярність в області інформаційно-комунікаційних технологій, його використання в індустрії Інтернету речей як і раніше є досить обмеженим. Специфікація AMQP описує такі функції, як орієнтація повідомлення, очікування в черзі, маршрутизація (включаючи обмін повідомленнями між точками і публікацію і підписку), надійність і безпеку. Ймовірно, найбільшою перевагою AMQP є його надійна комунікаційна модель. На відміну від MQTT, AMQP може гарантувати проведення повних транзакцій, що, хоча і корисно, але не завжди так, як того вимагають застосунки IoT.

Через свою важкість AMQP не підходить для сенсорних пристроїв з обмеженою пам'яттю, потужністю або пропускною спроможністю мережі, однак для окремих випадків використання IoT він може бути єдиним протоколом, придатним для наскрізного застосування, включаючи такі приклади, як промислове важке обладнання або системи SCADA, де пристрої та мережа, як правило, володіють значно більшими можливостями.

1.7.6 Легкий M2M (LwM2M)

Відмінність LwM2M від інших протоколів, що застосовуються в IoT, полягає в тому, що він був спеціально розроблений для задоволення вимог комплексної обробки обмежених ресурсів пристроїв. Запущена в 2014 році Open Mobile Alliance (нині OMA SpecWorks), вона забезпечує чітко визначений стандарт для передачі даних IoT і управління пристроями.

На відміну від традиційних M2M-рішень, в яких пристрою зазвичай потрібно підтримувати кілька стеків технологій, протоколів і служб безпеки, модель Lightweight M2M дозволяє користувачам мати один стек технологій для управління пристроєм не тільки на рівні самого пристрою, але і на рівні застосунків. Більш того, LwM2M пропонує крос-платформену сумісність, що робить її ідеальною для постачальників послуг, які бажають уникнути блокування постачальника. Поєднання DTLS, CoAP, Block, Observe, SenML LwM2M і Resource Directory, використовує їх для формування інтерфейсу пристрою-сервера з певною структурою об'єкта. З урахуванням всіх перерахованих вище переваг разом узятих, Lightweight M2M здатний забезпечити ідеальний час виведення на ринок, оскільки він доступний для миттєвого розгортання.

Ще одна сильна сторона LwM2M, що відрізняє його від інших протоколів M2M, доступних на ринку, полягає в тому, як він справляється з проблемами безпеки, особливо на пристроях з обмеженими ресурсами. Він заснований на розширеному протоколі DTLS, який підтримує облікові дані на основі спільно використовуваних ключів, необроблених відкритих ключів або сертифікатів і реалізує аутентифікацію, конфіденційність і цілісність даних між сервером і клієнтом.

Протокол Lightweight M2M пропонує чітко визначену структуру пристрою і моделі управління даними, що дозволяє використовувати ряд діагностичних функцій виробника, таких як безпечне переривання

завантаження пристрою, доступ до об'єктів або ресурсів, а також звітність пристрою.

Підводячи підсумок, можна сказати, що протокол Lightweight M2M пропонує гнучке, масштабоване діагностичне управління пристроями від виробника з поліпшеним часом виведення на ринок, що робить його особливо підходящим для малопотужних пристроїв з обмеженими можливостями обробки і зберігання. З огляду на все це, LwM2M є кращим рішенням для великого, складного і тривалого розгортання з використанням крос-платформних і стандартних IoT-сервісів.

1.8 Автентифікація як механізм безпеки IoT

Автентифікація - це процес визначення того, чи є хтось або щось насправді тим, за кого або ким він себе видає. Технологія аутентифікації забезпечує контроль доступу для систем шляхом перевірки відповідності облікових даних користувача облікових даних в базі даних авторизованих користувачів або на сервері аутентифікації даних.

Користувачі зазвичай ідентифікуються за допомогою ідентифікатора користувача, і аутентифікація здійснюється, коли користувач надає облікові дані, наприклад, пароль, який відповідає цьому ідентифікатору користувача. Більшість користувачів найкраще знайомі з використанням пароля, який, як частина інформації, яка повинна бути відома тільки користувачеві, називається фактором аутентифікації знань.

Автентифікація важлива, оскільки вона дозволяє організаціям підтримувати безпеку своїх мереж, дозволяючи лише аутентифікованим користувачам (або процесам) отримувати доступ до своїх захищених ресурсів, які можуть включати комп'ютерні системи, мережі, бази даних, веб-сайти і інші мережеві застосунки або послуги.

Після аутентифікації користувач або процес, як правило, також проходить процедуру авторизації, щоб визначити, чи повинен аутентіфікований суб'єкт мати доступ до захищеного ресурсу або системи. Користувач може бути аутентифікований, але йому не буде надано доступ до ресурсу, якщо йому не було дано дозвіл на доступ до нього.

Терміни "аутентифікація" і "авторизація" часто використовуються як взаємозамінні; хоча вони часто можуть бути реалізовані разом, ці дві функції є різними. У той час як аутентифікація - це процес перевірки особистості зареєстрованого користувача перед наданням доступу до захищеного ресурсу, авторизація - це процес підтвердження того, що аутентифікованому користувачеві було дано дозвіл на доступ до запитуваних ресурсів. Процес, за допомогою якого доступ до цих ресурсів обмежений певною кількістю користувачів, називається контролем доступу. Процес аутентифікації завжди передує процесу авторизації.

Аутентифікація користувачів відбувається в рамках більшості взаємодій між людьми і комп'ютерами. Як правило, користувач повинен вибрати ім'я користувача або ідентифікатор користувача і надати дійсний пароль, щоб почати користуватися системою. Аутентифікація користувачів дозволяє взаємодію між людьми і машинами в операційних системах і застосунках, а також в дротових і бездротових мережах для забезпечення доступу до мережевих і підключених до Інтернету систем, застосунків і ресурсів.

Багато компаній використовують аутентифікацію для валідації користувачів, що заходять на їх веб-сайти. Без належних заходів безпеки дані користувачів, такі як номери кредитних і дебетових карт, а також номери соціального страхування, можуть потрапити в руки кіберзлочинців.

Організації також використовують аутентифікацію для контролю того, які користувачі мають доступ до корпоративних мереж і ресурсів, а також для ідентифікації та контролю того, які машини і сервери мають доступ. Компанії також використовують аутентифікацію для забезпечення безпечного доступу віддалених співробітників до своїх програм і мереж.

Для підприємств та інших великих організацій аутентифікація може бути виконана за допомогою системи єдиного входу (SSO), яка надає доступ до декількох систем з одним набором облікових даних для входу.

Аутентифікація користувача з ідентифікатором користувача і паролем зазвичай вважається найпростішим способом аутентифікації і залежить від знання користувачем двох частин інформації: ідентифікатора користувача або імені користувача та пароля. Оскільки цей тип аутентифікації ґрунтується тільки на одному факторі аутентифікації, він є різновидом однофакторної аутентифікації.

Сильна аутентифікація - це термін, який не був офіційно визначений, але зазвичай використовується для позначення того, що використовуваний тип аутентифікації є більш надійним і стійким до атак; для досягнення цього зазвичай визнається, що для цього необхідно використовувати принаймні два різних типи факторів аутентифікації.

Фактор аутентифікації є деякою інформацією або атрибутом, які можуть бути використані для аутентифікації користувача, що запитує доступ до системи. Стара приказка про безпеку говорить, що факторами аутентифікації можуть бути "щось, що ви знаєте, щось, що у вас є або щось, що ви є". Ці три чинники відповідають фактору знань, фактору володіння і фактору неузгодженості. В останні роки були запропоновані і введені в дію додаткові фактори, причому в багатьох випадках в якості четвертого фактора виступає розташування, а в якості п'ятого фактора - час.

Традиційна аутентифікація залежить від використання файлу паролів, в якому зберігаються ідентифікатори користувачів разом з хешами паролів, пов'язаних з кожним користувачем. При вході в систему пароль, присланий користувачем, хешується і порівнюється зі значенням в файлі пароля. Якщо два хеша збігаються, користувач аутентифікований .

Такий підхід до аутентифікації має кілька недоліків, особливо щодо ресурсів, що використовуються в різних системах. По-перше, зловмисники, які мають доступ до файлу паролів для системи, можуть використовувати атаки

брутфорсу проти хешованих паролів для їх визначення. З іншого боку, такий підхід зажадає множинної аутентифікації для сучасних застосунків, що отримують доступ до ресурсів через багато систем.

Слабкі сторони аутентифікації на основі паролів можуть бути певною мірою усунені за допомогою розумніших імен користувачів і правил паролів, таких як мінімальна довжина і вимоги до складності, наприклад, включення заголовних букв і символів. Однак аутентифікація на основі пароля і аутентифікація на основі знань є більш уразливими, ніж системи, що вимагають декількох незалежних методів.

1.9 Детектор аномалій як механізм безпеки IoT

Виявлення аномалій - це процес ідентифікації несподіваних елементів або подій в наборах даних, які відрізняються від норми. Виявлення аномалій має два основних припущення:

Аномалії в даних зустрічаються дуже рідко. Їх характеристики істотно відрізняються від звичайних примірників.

Аномалії можна поділити на наступні категорії:

Точечні аномалії: Один екземпляр даних є аномальним, якщо він знаходиться занадто далеко від інших. Приклад: Виявлення шахрайства з кредитними картами на підставі "витраченої суми".

Контекстуальні аномалії: Аномалія залежить від конкретного контексту. Цей тип аномалій широко поширений в даних часових рядів. Приклад: Витрати в розмірі $ 100 кожен день в святковий сезон - це нормально, але може здатися дивним.

Колективні аномалії: Набір екземплярів даних в сукупності допомагає у виявленні аномалій. Приклад: Хтось несподівано намагається скопіювати дані з віддаленої машини на локальний хост, що є аномалією, яка буде позначена як потенційна кібератака.

Виявлення аномалій аналогічно, але не повністю збігається з виявленням шумів і новинок. Виявлення новинок пов'язано з виявленням небаченої закономірності в нових спостереженнях, які не включені в дані тренінгу, наприклад, раптовий інтерес до нового каналу на YouTube в Різдво. Видалення шуму (УЗШ) - це процес імунізації аналізу від небажаних спостережень, іншими словами, видалення шуму з сигналу, який в іншому випадку був би значущим.

1.9.1 Популярні види детекторів аномалій

Виявлення аномалій на основі щільності базується на алгоритмі k-найближчих сусідів.

Припущення: Звичайні точки збору даних розташовуються навколо щільного району, а аномалії - далеко.

Найближчий набір точок даних оцінюється з використанням балів, які можуть бути Евклідовою відстанню або схожою в залежності від типу даних (категорії або числа). Їх можна розділити на два алгоритми:

1. K-найближчий сусід: k-NN - це простий, непараметричний метод ледачого навчання, який використовується для класифікації даних на основі подібності в дистанційних метриках, таких як Евклідова, Манхеттова, Мінковська або Хаммінгова відстань.
2. Відносна щільність даних: Більш відома як місцевий фактор викиду (LOF). Ця концепція заснована на метриці відстані, званої відстанню досяжності.

Кластеризація є однією з найпопулярніших концепцій в області неконтрольованого навчання.

Припущення: Аналогічні точки даних, як правило, належать до аналогічних груп або кластерів, що визначаються їх віддаленістю від місцевих Центроїд.

K-засіб - це широко використовуваний алгоритм кластеризації. Він створює 'k' схожі кластери точок даних. Випадки, що виходять за рамки цих груп, можуть бути потенційно відзначені як аномалії.

Машина з вектором підтримки - ще один ефективний метод виявлення аномалій. SVM зазвичай асоціюється з контрольованим навчанням, але існують розширення (наприклад, OneClassCVM), які можуть бути використані для виявлення аномалій як неконтрольованих проблем (в яких дані навчання не позначені). Алгоритм вивчає межу для угруповання звичайних екземплярів даних за допомогою навчального набору, а потім, використовуючи тестовий екземпляр, сам налаштовується на виявлення аномалій, які виходять за межі досліджуваного регіону.

Залежно від конкретного випадку використання, виходом детектора аномалій можуть бути числові скалярні значення для фільтрації по порогам домену або текстовим міткам (наприклад, двійкові / мультиметрові мітки).

Висновки до розділу 1

Протягом останніх двох десятиліть Інтернет речей продовжує стрімко розширюватися по всьому світу. Подолавши шлях у багато галузей промисловості, такі як виробництво, охорону здоров'я, автомобілебудування, безпеку, транспорт і багато іншого, він значно розширив можливості підприємств і приніс їм економічну вигоду.

Сьогодні "Інтернет речей" підтримує десятки різних протоколів Інтернету речей. У зв'язку з цим багато експертів IoT почали закликати до глобальної стандартизації протоколів. Однак, будучи фрагментованим за своєю природою, ринок Інтернету речей, ймовірно, ніколи не буде мати потребу у всеосяжному стандарті. Так само, як з'являються все нові і нові застосукнки і випадки використання в індустрії Інтернету речей, які підходять для конкретних цілей, протоколи Інтернету речей будуть з'являтися і надалі. Слід ще раз підкреслити,

що безпечне та ефективне управління пристроями є наріжним каменем стійкого розвитку мереж Інтернету речей у всьому світі. Це одна з причин, чому опис і осмислення різних протоколів IoT дійсно важливий. Тому насправді необхідно знати потреби і вимоги свого бізнесу, знати про переваги і недоліки протоколів, пропонованих на ринку, і вміти вибирати протокол, який найкраще підходить для конкретного випадку використання. Ринок IoT пристроїв не так давно почав розвиватися і приносити нові загрози в інформаційні мережі, а наслідки, створені атаками, можуть бути непередбачуваними. Вирішено проаналізувати стан безпеки IoT, що і буде зроблено в наступному розділі.

2 АНАЛІЗ ПРОБЛЕМ БЕЗПЕКИ В IOT

2.1 Аспекти безпеки

2.1.1 Питання безпеки IoT

Використання сполучених об'єктів в повсякденному житті людей може зробити питання безпеки життя дуже актуальним. Інтегрована в будинку, автомобілі та електромережі інтелектуальність може бути використана хакерами для створення небезпечних ситуацій. Різні сценарії злому, представлені в останні роки, ілюструють рівень шкоди, яку може бути заподіяно порушеннями безпеки, особливо при розробці і широкому впровадженні застосунків IoT, що працюють з конфіденційною інформацією.

Основними проблемами безпеки Інтернету речей є: аутентифікація, авторизація, цілісність, конфіденційність, відмовостійкість, доступність і приватність.

1. Автентифікація: Процес підтвердження особистості об’єкту. В контексті Інтернету речей кожен об'єкт повинен мати здатність ідентифікувати і автентифікувати всі інші об'єкти в системі (або в певній частині системи, з якої він взаємодіє).
2. Авторизація: Процес видачі дозволу суб'єкту на здійсннення певних дій.
3. Цілісність: Шлях до забезпечення послідовності, точності і надійності інформації протягом усього її життєвого циклу. У IoT зміна базової інформації або навіть введення недійсної інформації може викликати серйозні проблеми, наприклад, у випадках використання інтелектуальних систем охорони здоров'я це може привести до смерті пацієнта.

4. Конфіденційність: Процес забезпечення доступу до інформації тільки уповноваженим особам. Відносно конфіденційності в IoT слід розглянути два основних питання: по-перше, забезпечення того, щоб об'єкт, який одержує дані, не переміщував / передавав ці дані іншим об'єктам, і, по-друге, вести управління даними.
5. Відмовостійкість: Спосіб гарантування здатності продемонструвати, що завдання або подія відбулася, з метою, що це не може бути спростовано пізніше. Іншими словами, об'єкт не може заперечувати справжність переданих даних.
6. Доступність: Процес забезпечення доступності необхідних послуг в будь-якому місці і в будь-який час для можливих користувачів. В IoT Це включає доступність об'єктів самих до себе.
7. Приватність: Процес забезпечення недоступності приватної інформації публічними або шкідливим об'єктами.

2.1.2 Чому безпека важлива для IoT

Безпека Інтернету речей важлива, оскільки багато критично важливих функцій відведено на підключені пристрої, а складні атаки можуть легко призвести до катастрофічних наслідків. Наприклад, в меншому масштабі хакери можуть отримати доступ в "розумний будинок", віддалено відключивши систему безпеки. У більш широкому масштабі, хакери можуть отримати контроль над комунальними мережами і відключити електрику в будівлі або навіть в районі.

Основна причина, по якій компанії зазнають труднощів із забезпеченням безпеки Інтернету речей, полягає в тому, що, прагнучи вивести пристрої Інтернету речей на ринок, виробники пристроїв Інтернету речей можуть

відмовитися від безпеки. Створення протоколів безпеки Інтернету речей з самого початку було б дорогою і трудомісткою справою, до того ж це могло б поставити під загрозу ті можливості, які найбільше потрібні споживачам. В результаті, компанії змушені мати справу з пристроями з меншою кількістю вбудованих міркувань безпеки.

Більшість пристроїв IoT мають аутентифікацію паролем і основні протоколи безпеки, але цього недостатньо. Оскільки пристрої IoT настільки спеціалізовані за розміром, масштабом і складності, багато стандартних рішень щодо забезпечення безпеки ПК працювати не будуть. Методи мережевої безпеки, з якими MSP і компанії найкраще знайомі - наприклад, брандмауери або програми для злому - розроблені для звичайних IT-інфраструктур, не обов'язкових IoT протоколів.

Кібербезпеку Інтернету речей також важко реалізувати за п'ятьма основними причинами:

1. Недостатньо ресурсів для створення надійної системи безпеки Інтернету речей: Підключені пристрої часто налаштовані на виконання одного процесу, і для забезпечення безпеки Інтернету речей не вистачає обчислювальної потужності.
2. "Установи і забудь про це": Пристрої Інтернету речей зазвичай не виправляються і не оновлюються після включення.
3. Відсутність встановлених стандартів безпеки Інтернету речей: Без формальної інфраструктури або фреймворків стандарти безпеки в пристроях IoT визначаються самими виробниками.
4. Надійність облікових даних за замовчуванням: Підключені пристрої працюють "з коробки", тільки якщо вони використовують стандартні облікові дані, які легко вгадуються хакерами. Аналогічним чином, пристрої IoT зазвичай виробляються масово - якщо ви можете зламати один пристрій, ви можете зламати їх все.

5. Тривалий термін служби виробу: IoT-пристрої знаходяться в обігу від 15 до 20 років. У зв'язку з таким тривалим терміном служби вони просто не зможуть йти в ногу зі стандартами безпеки без оновлень.

З цих причин пристрої Інтернету речей часто залишаються незахищеними і легко використовуються зловмисниками. Дослідники в області безпеки виявили, що кількість кібератак на пристрої Інтернету речей в 2019 році зросло до 2,9 млрд. Що в три рази більше, ніж в останні роки. Деякі з цих атак спрямовані проти самих пристроїв, проте кіберзлочинці, швидше за все, будуть спрямовані проти прогалин у безпеці пристроїв Інтернету речей, оскільки вони можуть використовуватися в якості точок входу в більші мережі.

2.1.3 Що варто зробити якщо ви маєте IoT пристрій

Основні функціональні рішення щодо забезпечення безпеки Інтернету речей повинні полягати в захисті пристроїв Інтернету речей від витоків даних і кібер атак, встановлення захищених комунікацій і забезпеченні захисту від несанкціонованого втручання в мікропрограмму.

Безпека пристроїв Інтернету речей може бути складним завданням, але є кілька речей, які кожен MSP повинен зробити для захисту конфіденційної інформації та зміцненню довіри клієнтів. Наступні кроки слід розглядати як основу кращих практик забезпечення безпеки Інтернету речей:

1. Створити окрему мережу: Зарезервуйте приватну мережу, доступ до якої можуть отримати тільки уповноважені співробітники.
2. Радьте клієнтам вибирати складні паролі: Під час налаштування протоколів безпеки для ваших клієнтів підкреслюйте важливість створення паролів, які хакеру буде важко вгадати або розшифрувати.
3. Утримайтеся від використання Universal Plug and Play (UPnP): Інструмент UPnP полегшує автоматичне виявлення пристроїв в мережі

і підключення до них. На жаль, хакери можуть підключатися до UPnP для отримання доступу до більш важливих пристроїв.

4. Регулярно відстежуйте і оцінюйте пристрої: Постійне відстеження місця розташування і стану всіх пристроїв Інтернету речей. Крім того, хоча тенденція IoT приваблива і має багато переваг, MSP і їх клієнти повинні ретельно обміркувати, скільки пристроїв насправді має бути підключено до IoT. Обмеження кількості використовуваних вами пристроїв обмежує можливості для атак зловмисникам.
5. Беріть участь в управлінні життєвим циклом безпеки Інтернету речей: MSP повинні думати про безпеку пристроїв Інтернету речей як про безперервний цикл. Заходи безпеки Інтернету речей повинні регулярно вивчатися, впроваджуватися, оновлюватися і аналізуватися, щоб не відставати від постійно змінюючихся загроз.

MSP також повинні враховувати ризик атак, супутні витрати, пов'язані зі збоями в системі безпеки Інтернету речей, і початкові витрати на впровадження рішення по забезпеченню мережевої безпеки Інтернету речей в процесі планування.

2.1.4 Наскільки безпечні IoT пристрої

Пристрої Інтернету речей так само безпечні, як і їх виробники. Всі технології певною мірою уразливі для кібератак, тому було б несправедливо змальовувати пристрої Інтернету речей як менш безпечні, ніж їх колеги.

При цьому пристрої Інтернету речей схильні до вищого ризику злому, ніж смартфони або комп'ютери. Хоча підключені пристрої часто передають конфіденційну інформацію, звичайні користувачі не вважають свої інтелектуальні пристрої загрозою безпеки.

Коли постачальники інтернет-послуг висувають безпеку Інтернету речей на перший план у своїй клієнтській стратегії, їх клієнти можуть з упевненістю

зосередитися на тому, для чого IoT було створено - для аналізу більш точних даних, поліпшення загальної якості обслуговування і підвищення залученості клієнтів.

2.1.5 Заходи безпеки, які можна прийняти для забезпечення безпеки пристроїв

Технології Інтернету речей становлять потенційну небезпеку для вашої безпеки в Інтернеті. Повідомлення про новини варіювалися від ботнету Інтернету речей, що відключав частини Інтернету, до хакерів, що використовують радіонянь.

Ось чому є хорошою ідеєю захистити ваше цифрове життя, захищаючи ваші пристрої, підключені до IoT. Дев’ять способів зробити це:

1. Встановіть на комп'ютери, планшети і смартфони надійне програмне забезпечення для забезпечення безпеки в Інтернеті. Наприклад, Norton Security Deluxe може забезпечити захист в режимі реального часу від існуючих і з'являючихся шкідливих програм, включаючи програми-викупи і віруси.
2. Використовуйте надійні і унікальні паролі для облікових записів пристроїв, мереж Wi-Fi і підключених пристроїв. Не використовуйте поширені слова або паролі, які легко вгадати, такі як "пароль" або "123456".
3. Будьте уважні, коли справа доходить до застосунків. Обов'язково ознайомтеся з політикою конфіденційності застосунків, які ви використовуєте, щоб дізнатися, як вони планують використовувати вашу інформацію і багато іншого.
4. Проведіть свої дослідження, перш ніж купувати. Пристрої стають інтелектуальними, тому що вони збирають велику кількість персональних даних. Хоча збір даних не обов'язково є поганою річчю,

ви повинні знати, які типи даних збирають ці пристрої, як вони зберігаються і захищаються, чи надаються дані третім особам, і які політики або засоби захисту від витоку даних.

5. Дізнайтеся, до яких даних пристрій або застосунок хоче отримати доступ на телефоні. Якщо це здається зайвим для функціональності програми або занадто ризикованим, відмовте в дозволі.
6. Використовуйте VPN, як Norton Secure VPN, який допомагає захистити дані, що передаються через ваш будинок або громадський Wi-Fi.
7. Регулярно перевіряйте оновлення прошивки на веб-сайті виробника пристрою.
8. Будьте обережні при використанні функцій соціального обміну з цими застосунками. Функції соціального обміну можуть відкрити доступ до такої інформації, як ваше місце розташування, і дати людям знати, коли ви перебуваєте поза домом. Кіберзлочинці можуть використовувати це для відстеження ваших переміщень. Це може привести до потенційної проблеми кіберпереслідування або іншим реальним загрозам.
9. Ніколи не залишайте свій смартфон без нагляду, якщо ви використовуєте його в громадських місцях. У людних місцях слід також відключити доступ Wi-Fi або Bluetooth, якщо вони вам не потрібні. Деякі бренди смартфонів дозволяють автоматично обмінюватися інформацією з іншими користувачами в безпосередній близькості.

2.1.6 Безпека кінцевих точок Інтернету речей в порівнянні з мережевою безпекою

Безпека пристроїв Інтернету речей - справжня проблема. Пристрої Інтернету речей відрізняються високим ступенем диверсифікації та оснащені широким спектром операційних систем (операційних систем реального часу, на базі Linux або порожніх), комунікаційних протоколів і архітектури. На додаток до великої різноманітності, виникають питання нестачі ресурсів і відсутності галузевих стандартів і правил. Більшість рішень безпеки сьогодні фокусуються на захист мережі (виявляють мережеві аномалії і домагаються видимості активних в мережі пристроїв IoT), в той час як розуміння того, що самі пристрої повинні бути захищені, тільки посилюється. Той факт, що пристрої IoT можуть бути легко використані, робить їх дуже хорошою мішенню для зловмисників, які прагнуть використовувати слабкі пристрої IoT в якості точки входу на всю мережу підприємства, не потрапляючи в неї. Крім того, важливо пам'ятати, що мережеві рішення не мають відношення до розподілених IoT-пристроїв, які не мають мережі для їх захисту.

Тому виробники пристроїв IoT грають ключову роль в забезпеченні безпеки середовища IoT, і все більше організацій готові платити більше і більше за вбудований захист своїх інтелектуальних пристроїв.

2.1.7 Вразливості Інтернету речей сторонніх виробників

Однією з основних проблем в безпеці Інтернету речей є сильна залежність пристроїв Інтернету речей від сторонніх компонентів для забезпечення комунікаційних можливостей, криптографічних можливостей, самої операційної системи і т.д. Насправді, ця залежність настільки велика, що вона досягла такого ступеня, що навряд чи можна знайти пристрій IoT без сторонніх компонентів всередині нього. Той факт, що сторонні бібліотеки

широко використовуються на різних пристроях, в поєднанні з труднощами в їх захисті, робить їх кращим місцем для хакерів, щоб знайти уразливості Інтернету речей і використовувати багато пристроїв Інтернету речей через такі сторонні компоненти.

Вразливість в компонентах сторонніх виробників дуже небезпечна. У багатьох пристроях IoT не існує поділу і сегментації між процесами і / або завданнями, що означає, що навіть одна вразливість в сторонній бібліотеці ставить під загрозу весь пристрій. Це може привести до смертельних результатів: зловмисники можуть використовувати вразливість третьої сторони для захоплення контролю над пристроєм і заподіяння шкоди, крадіжки інформації про виконання атаки програми-викупу на виробника.

Компоненти сторонніх виробників небезпечні, і їх також надзвичайно складно убезпечити. Багато компонентів сторонніх виробників поставляються в двійковій формі, без доступу до вихідного коду. Навіть коли вихідний код доступний, часто буває важко зануритися в нього і оцінити рівень безпеки або уразливості всередині нього. У будь-якому випадку, більшість розробників використовують компоненти з відкритим вихідним кодом в якості чорних ящиків. Крім того, інструменти статичного аналізу та прапори безпеки компілятора не мають можливості аналізувати і захищати сторонні компоненти, а більшість рішень з безпеки IoT не можуть забезпечити захист двійкового коду в режимі реального часу.

2.2 Атаки на автентифікацію в IoT

У цьому розділі ми розглянемо найбільш базову архітектуру IoT (трьохрівневу архітектуру) і обговоримо проблеми безпеки, атаки і вимоги безпеки на кожному рівні архітектури. Зокрема атаки на автентифікацію.

2.2.1 Питання і вимоги до безпеки на рівні сприйняття

Рівень сприйняття складається з датчиків, які характеризуються обмеженою обчислювальною потужністю і ємністю. У зв'язку з такими обмеженнями виникає ряд проблем безпеки і ризиків атак.

Звернемо увагу на кілька атак на рівень сприйняття:

1. Захоплення вузла: Вузли можуть легко контролюватися зловмисниками. Захоплення вузла дозволяє противнику не тільки отримати доступ до криптографічних ключів і станів протоколів, але і до клонування і перерозподілу шкідливих вузлів в мережі, що впливає на безпеку всієї мережі.
2. Атака відмови в обслуговуванні (DoS): Тип атак, який відключає систему або мережу і не дозволяє авторизованим користувачам отримати до неї доступ. Це може бути досягнуто шляхом одночасного перевантаження системи або мережі великою кількістю запитів на розсилку спаму, що призводить до перевантаження системи і перешкоджає їй надавати нормальні послуги.
3. Атака відмови у сні: Однією з основних завдань мережі IoT є здатність зондування через велику кількість розподілених вузлів, кожен з яких надає невеликі дані, такі як температура, вологість, вібрація і т.д., в заданий інтервал часу, а потім їх перехід в сплячий режим ще на один інтервал часу, що дозволяє вузлам працювати протягом тривалого терміну служби. Атака типу "відмова уві сні" працює на джерело живлення вузла з метою збільшення енергоспоживання для скорочення терміну служби вузла шляхом запобігання засипання вузла після відправки відповідних сенсорних даних.

4. Розподілена атака типу «відмова в обслуговуванні" (DDoS): Широкомасштабний варіант DoS-атак. Найбільш складною проблемою є можливість використання великої кількості вузлів IoT для передачі трафіку, зібраного на сервер жертву. Є ознаки того, що DDoS-атака під назвою "Мірай", що відбулася в жовтні 2016 року, одержала успіх від великої кількості вузлів Інтернету речей.
5. Підроблений вузол: Тип атак, при яких зловмисник може використовувати підроблені ідентифікаційні дані, використовуючи підроблені вузли. При наявності ураженого вузла вся система може генерувати невірні дані або навіть сусідні вузли можуть отримувати спам і отримати порушення конфіденційності. Підроблені вузли можуть бути використані для передачі даних на "легітимні" вузли, що призводить до споживання ними енергії, що може привести до зниження якості всієї послуги.
6. Атака Відтворення: В ході цієї атаки інформація зберігається і передається повторно, не маючи на те відповідних повноважень. Такі атаки зазвичай використовуються проти протоколів аутентифікації.
7. Маршрутизація загроз: Цей тип атак є найбільш фундаментальною атакою на мережевому рівні, але може відбуватися на рівні сприйняття в процесі переадресації даних. Зловмисник може створити цикл маршрутизації, що приводить до браку або розширенню маршруту, збільшення затримки і збільшення кількості повідомлень про помилки.
8. Атака побічного каналу: Цей тип атак відбувається на шифрувальні пристрої, використовуючи апаратну інформацію, на яку накладається криптосистема (чіпи), таку як час виконання, енергоспоживання, розсіювання потужності і

електромагнітні перешкоди, вироблені електронними пристроями в процесі шифрування. Така інформація може бути проаналізована для виявлення секретних ключів, які використовуються в процесі шифрування.

9. Масова аутентифікація вузла: Процес аутентифікації великої кількості пристроїв в системі IoT, який вимагає великої кількості мережевих з'єднань для завершення етапу аутентифікації, що може вплинути на продуктивність всієї системи.

Беручи до уваги вищевказані ризики, існує необхідність в аутентифікації вузлів для запобігання підроблених вузлів і незаконного доступу, а також в шифруванні даних для захисту конфіденційності даних при передачі між вузлами (кінцевий вузол, шлюз або сервер). У зв'язку з властивостями вузлів, пов'язаними з нестачею електроенергії і обмеженою ємністю сховища, існує необхідність в зрілих легких схемах безпеки, які включають в себе як легкі криптографічні алгоритми, так і протоколи безпеки.

2.2.2 Питання та вимоги безпеки на мережевому рівні

Мережевий рівень відповідає за поширення даних з рівня сприйняття на прикладний рівень. Саме тут відбувається маршрутизація даних, а також первинний аналіз даних. На цьому рівні використовується кілька мережевих технологій, таких як різні технології для мобільних поколінь зв'язку (2G, 3G, 4G і 5G) і бездротових мереж (Bluetooth, WiMAX, WiFi, LoRaWAN і т.д.).

Виявлено кілька атак і ризиків на мережевому рівні:

1. "Людина посередині" (MITM): Згідно McAfee, найбільш частими атаками є атаки типу "відмова в обслуговуванні" (DoS) і "людина в браузері" (MITB). Останнє, поряд з атакою на рівні захищених сокетів (SSL), яка дозволяє зловмисникам прослуховувати трафік,

перехоплювати його і підробляти обидва кінці даних, становить MITM-атаку.

2. Відмова в обслуговуванні (DoS): Цей тип атак відбувається також на мережевому рівні, блокуючи передачу радіосигналів, використовуючи підроблений вузол, впливаючи на передачу або маршрутизацію даних між вузлами.
3. Підслуховування / спостерігання: Цей тип пасивних атак дає зловмиснику можливість прослуховувати приватну переписку по каналу зв'язку. Зловмисник може витягнути корисну інформацію, таку як імена користувачів і паролі, ідентифікація або конфігурація вузлів, які можуть привести до інших типів атак, наприклад, підробленим вузлам, атакам повторів і т.д.
4. Маршрутні атаки: Цей тип атак впливає на маршрутизацію повідомлень або даних. Зловмисник підміняє, перенаправляє, переадресовує на неправельне джерело або навіть зкидує пакети на мережевому рівні. Можна розглянути наступні конкретні напади:
    - Чорна діра: Також можна розглядати як DoS-атаку, в якій зловмисник використовує підроблений вузол, який вітає весь трафік, стверджуючи, що у нього найкоротший шлях. В результаті весь трафік перенаправлятиметься на підроблений вузол, який має можливість перенаправляти його на проксі-сервер або навіть скидати його.
    - Сіра діра: цей тип атак схожий на атаку чорної діри, але замість того, щоб скинути всі пакети, він скине тільки обрані.
    - Червоточина: В цьому типі атак зловмисник створює з'єднання між двома точками в мережі, контролюючи принаймні два вузла мережі або додаючи нові підроблені вузли в мережу. Після формування посилання зловмисник збирає дані з одного кінця і відтворює їх на інший кінець.

- Флуд вітань: Метою зловмисника в цьому типі атак є використання потужності вузлів системи шляхом трансляції пакетів Hello запиту фальшивим вузлом, щоб вплинути на всі вузли системи, що знаходяться в одному діапазоні, що призводить до відправки кожним з них пакетів на сусідів, викликаючи величезний трафік в мережі.
- Сібіл: У цій атаці, фальшивий вузол видає декілька ідентичностей, таким чином, він може контролювати значну частину структури, перебуваючи в різних місцях в мережі в один і той же час. Коли багато вузлів Sybil знаходяться в одній і тій же мережі, вони будуть посилати велику кількість інформації, що заблокує використання мережі звичайними вузлами.

Ці потенційні атаки на мережевому рівні (дротовому або бездротовому) призводять до визначення таких вимог безпеки: hop-to-hop шифрування, point-to-point аутентифікація, узгодження ключів і управління ними, маршрутизація безпеки і виявлення вторгнень.

2.2.3 Питання і вимоги безпеки на рівні додатків

Рівень додатків відповідає за надання послуг. Він містить набір протоколів для передачі повідомлень, наприклад, Constrained Application Protocol (COAP), Message Queuing Telemetry Transport (MQTT), Extensible Messaging and Presence Protocol (XMPP), Advanced Message Queuing Protocol (AMQP) і т.д. Цей рівень безпосередньо взаємодіє з користувачем. З огляду на те, що "традиційні" протоколи рівня додатку не дуже добре працюють в IoT, і оскільки IoT не має власних міжнародних стандартів, на прикладному рівні виникає кілька проблем безпеки.

1. Доступність і аутентифікація даних: Кожна програма може мати багато користувачів. Фальшиві або нелегальні користувачі можуть спричинити великий вплив на доступність всієї системи. Така велика кількість користувачів означає різні права і контроль доступу.
2. Конфіденційність і справжність даних: Той факт, що IoT з'єднує різні пристрої різних виробників, призводить до застосування різних схем аутентифікації. Інтеграція цих схем є складним завданням для забезпечення конфіденційності та справжності даних.
3. Робота з наявністю великих масивів даних: IoT з'єднує величезну кількість кінцевих пристроїв, що призводить до необхідності управління величезною кількістю даних. Це призводить до додаткових витрат додатком для аналізу цих даних, що має великий вплив на доступність сервісів, що надаються додатком.

Що стосується вимог безпеки на прикладному рівні, аутентифікація необхідна при захисті конфіденційності користувачів (відповідно, даних). Також повинна існувати схема управління інформаційною безпекою, яка включає управління ресурсами і управління інформацією з фізичної безпеки. На Рисунку 2.1 наведена зведена інформація про вимоги безпеки трирівневої архітектури Інтернету речей.

Layer	Security Requirements
Perception	Lightweight Encryption
	Authentication
	Key Agreement
	Data Confidentiality
Network	Communication Security
	Routing Security
	Authentication
	Key Management
	Intrusion Detection
Application	Authentication
	Privacy protection
	Information Security Management

Рисунок 2.1 – Архітектурні вимоги безпеки

По Рисунку 2.1 ясно, що аутентифікація є основним механізмом безпеки, який повинен застосовуватися на різних рівнях. У разі використання Інтернету речей може знадобитися аутентифікація між кінцевими пристроями і проміжним пристроєм (шлюзом). Шлюз повинен аутентифіковуватися під час відправки даних в хмару, а додаток (мобільний або веб) має бути аутентифікований в хмару для збору даних з метою аналізу.

2.3 Таксономія схем аутентифікації Інтернету речей

В цьому розділі подано класифікацію схем аутентифікації IoT з використанням різних критеріїв, обраних на основі подібності і основних характеристик цих схем. Як згадувалося вище, аутентифікація може застосовуватися на кожному з трьох рівнів архітектури IoT, що робить методи аутентифікації різноманітними. Ці критерії проілюстровані на Рисунку 2.2 і коротко викладені нижче.

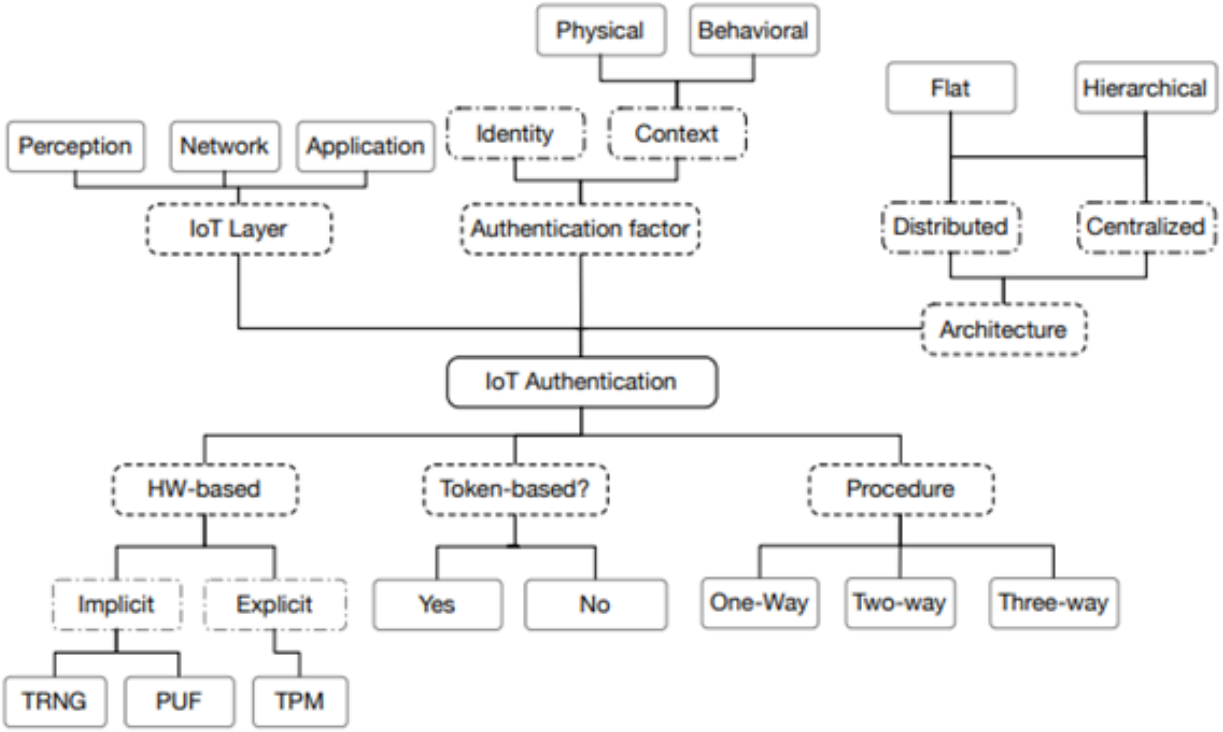

Рисунок 2.2 - Таксономія схем аутентифікації IoT

2.3.1 Фактори аутентифікації

- На основі ідентифікації: Інформація, представлена однією стороною іншій для підтвердження своєї автентичності. Схеми аутентифікації на основі ідентифікації можуть використовувати один (або комбінацію) хеш, симетричний або асиметричний криптографічний алгоритм.

- Контекстний: який може бути:

  - Фізичний: Біометрична інформація, заснована на фізичних характеристиках людини, наприклад відбитки пальців, геометрію руки, сканування сітківки ока і т.д.
  - Поведінковий: Біометричні дані, засновані на поведінкових характеристиках людини, наприклад, динаміка натискання клавіш (характер ритму і часу, що створюються при наборі тексту), аналіз ходи (метод, який використовується для

оцінки того, як ми ходимо або біжимо), голосовий ID (голос) аутентифікація з використанням голосового друку) і т.д.

2.3.2 Використання токенів

- Аутентифікація на основі токена: аутентифікація користувача / пристрою на основі ідентифікаційного токена (фрагмента даних), створеного сервером, таким як протокол OAuth2 або відкритий ID.

- Безтокенова аутентифікація: Включає використання облікових даних (ім'я користувача / пароль) щоразу, коли виникає необхідність в обміні даними (наприклад, TLS / DTLS).

2.3.3 Процедура аутентифікації

- Одностороння аутентифікація: У разі, коли дві сторони бажають спілкуватися один з одним, тільки одна сторона аутентифіковує себе іншій, в той час як інша залишається не аутентифікованою.
- Двостороння аутентифікація: Також називається взаємною аутентифікацією, при якій обидва суб'єкта аутентифіковують один одного.
- Тристороння аутентифікація: Коли центральний орган аутентифікує обидві сторони і допомагає їм взаємно аутентифицировать себе.

2.3.4 Архітектура аутентифікації

- Розподілена: Використання методу розподіленої прямої аутентифікації між сторонами.

- Централізована: Використання централізованого сервера або довіреної третьої сторони для розподілу і управління обліковими даними, що використовуються для аутентифікації.

Незалежно від того, централізована вона або розподілена, архітектура схеми аутентифікації може бути такою:

- Ієрархічна: використання багаторівневої архітектури для обробки процедури аутентифікації.
- Пласка: для процедури аутентифікації не використовується ієрархічна архітектура.

2.3.5 Рівні IoT

Вказує на рівень, на якому застосовується процедура аутентифікації.

- Рівень сприйняття: Відповідає за збір, обробку та оцифровку інформації, яка сприймається кінцевими вузлами платформи IoT.
- Мережевий рівень відповідає за отримання даних з рівня сприйняття і їх обробку.
- Рівень додатків: Відповідає за отримання даних з мережевого рівня, а потім за надання послуг, запитуваних користувачами.

2.3.6 Апаратні засоби

Процес аутентифікації може потребувати використання фізичних характеристик апаратного забезпечення або самого апаратного забезпечення.

- Неявні апаратна основа: Використовують фізичні характеристики апаратного забезпечення для поліпшення аутентифікації, такі як фізична неповторювана функція (PUF) або генератор справжніх випадкових чисел (TRNG).

- Явно виражена апаратна основа: Деякі схеми аутентифікації засновані на використанні модуля довіреної платформи (TPM), чіпа (апаратного забезпечення), який зберігає і обробляє ключі, які використовуються для апаратної аутентифікації.

2.4 Моделі виявлення аномалій для даних часових рядів IoT

Алгоритми для виявлення аномалій в даних часових рядів датчиків можна розділити на наступні макрокласси:

- Статистичні методи: ці методи використовують минулі вимірювання для апроксимації моделі правильної поведінки датчика (або будь-якого компонента, за яким ми намагаємося спостерігати).

  Кожен раз, коли реєструється новий вимір, він порівнюється з моделлю і, якщо він статистично несумісний з нею, то позначається як аномалія. Статистичні методи можуть застосовуватися не тільки до одиничних показників, але і до вікон показань. Зазвичай віконний підхід допомагає скоротити кількість помилкових спрацьовувань. Прикладом дуже поширеного статистичного методу виявлення аномалій є так званий фільтр низьких частот, який класифікує показання як аномалії в залежності від того, наскільки вони відрізняються від середніх результатів минулих вимірів.
- Імовірнісні методи: ці методи обертаються навколо визначення ймовірнісної моделі, яка може бути параметричною або непараметричною (в залежності від того, відповідають виміри датчиків добре відомому розподілу чи ні). Потім проводиться класифікація

аномалій шляхом вимірювання ймовірності зчитування по відношенню до моделі. Якщо ймовірність опускається нижче визначеного порога, то вона позначається як аномальна подія. Моделі, виявлені ймовірносними методами, можуть бути дуже простими, але також і дуже складними, можливо, кодуючими відносини між вимірами в часі, використовуючи або приховані марковские моделі (HMM), або Байєсовські мережі (BNs).

Однак методи, засновані на BN і HMM, зазвичай дуже дорогі в обчисленнях і погано масштабуються, особливо при потоковій передачі даних.

- Методи, засновані на наближенні: ці методи засновані на відстанях між вимірами даних для розмежування між аномальними і коректними показаннями. Дуже відомим алгоритмом, заснованим на проксимальних параметрах, є місцевий коефіцієнт викиду (LOF), який присвоює кожному r_i значення відхилення, засновані на щільності вимірювань навколо k найближчих сусідів і щільності вимірювань навколо r_i. Показання з високими балами відхилення відзначені як аномалії.
- Методи, засновані на кластеризації; ці методи є підмножиною алгоритмів, заснованих на близькості розташування. Тут вимірювання в першу чергу використовуються для створення кластерів. Потім нові виміри, які призначаються невеликим і ізольованим кластерам або вимірам, які знаходяться дуже далеко від центру їх кластера, позначені як аномальні.
- Методи, засновані на прогнозуванні: ці методи використовують минулі вимірювання для навчання моделі, яка може передбачити значення наступного

вимірювання в даних часових рядів датчиків. Якщо фактичний вимір занадто відрізняється від прогнозованого, то він позначається як аномальний. Існує безліч алгоритмів виявлення аномалій, заснованих на прогнозуванні, деякі з яких засновані на дуже простих моделях машинного навчання, таких як SVM 1 класу, а інші набагато більш складні і використовують в якості моделі прогнозування Deep Neural Networks (DNN) з Long Short-Term Memory (LSTM) компонентами як його модель передбачень.

Очевидно, що вибір алгоритму сильно залежить від типу даних, що підлягають моніторингу. Наприклад, кластерні підходи і підходи, засновані на принципі близькості розташування, погано працюють з об'ємними даними, оскільки відстані між точками, як правило, збільшуються у міру збільшення числа вимірювань.

Аналогічним чином, методи, засновані на HMMM і BN, не можуть ефективно обробляти багатовимірні дані і тому підходять тільки для одновимірних вибірок.

2.5 Приклад аналізу та попередньої обробки даних

Вибір того, який алгоритм використовувати, безсумнівно, вимагає певного рівня знань як про проблему, так і про дані. Для того, щоб зрозуміти, що може бути використано для виявлення аномалій в датчику аміаку, спочатку потрібно проаналізувати набір даних наданий з ETC Riccione.

Початковий набір даних складався з декількох CSV-файлів, кожен з яких містив одну годину вимірів. Проте, ми реорганізували його для отримання окремого CSV-файлу для кожного датчика (наприклад, один файл для всіх вимірів датчика аміаку на першій трасі, інший файл для датчика аміаку на другій трасі і

так далі). Кожен рядок цих нових CSV файлів містив показники датчиків, разом з міткою часу зняття виміру, ім'я датчика і цілочисельне значення. Знаття показань проводилися з похвилинною періодичністю, в цілому 1440 вимірів в день. Набір даних охоплює період з початку квітня 2016 року по початок липня 2017 року. [9]

2.5.1 Очищення даних

Ми почали з застосування порогового фільтра шуму для видалення показань несправних датчиків. Фільтр працював шляхом ідентифікації і видалення всіх вимірювань, які або були NaN, або проводилися дуже рідко (наприклад, значення порядку 1032), або не мали сенсу (наприклад, негативні значення).

Як тільки раптові сплески, NaNs і негативні значення були видалені, нам довелося зіткнутися з іншою проблемою, на цей раз з тимчасовими мітками придбань. Деякі мітки часу, по суті, не були унікальними, так як 31 жовтня перехід на стандартний час відсунув годинник на одну годину назад з 3:00 до 2:00 ранку. Це призводить до дублювання ключів у тимчасових рядах датчиків. Щоб вирішити цю проблему і зробити тимчасові ряди корисними, ми вирішили звести дві години вимірів після 2:00 ранку 31 жовтня в одну годину, видаливши повторювані тимчасові мітки.

Це було зроблено шляхом конденсації пар придбань, зроблених протягом двох хвилин поспіль ti і ti + 1, перетворюючи їх в одну, представлену їх середнім ½ (ti + ti + 1). [l]

2.5.2 Візуалізація даних

Після того, як набір даних був очищений, ми приступили до візуалізації даних. Ми побудували тимчасові ряди датчиків і їх розподіл значень і проаналізували їх. Ми розклали тимчасові ряди вимірів аміаку і розглянули їх динаміку, періодичність і залишковість. Ми також провели на ньому тест Діккі-Фуллера і виявили, що в ньому представлені стаціонарні властивості (повторювана поведінка по відношенню до календарного часу).

Точніше кажучи, ми виявили схожі закономірності в вимірах аміаку в порівнянні з аналогічним місяцем і сезоном. Це інформувало нас про те, що час збору даних може виявитися цінною додатковою функцією для нашого алгоритму виявлення аномалій.

2.6 Аутентифікація і контроль доступу в IoT

Безпека Інтернету речей є актуальною темою для дослідження, існує величезна кількість публікацій, що вказують на проблеми безпеки і конфіденційності в IoT. Через величезну кількість пристроїв IoT і можливості взаємодії "машина-машина" в IoT, традиційні методи аутентифікації і авторизації для нього нежиттєздатні. Пристрої повинні аутентифіковуватися один у одного перед обміном будь-якою інформацією (M2M комунікація), що є проблемою для дослідника через велику кількость пристроїв.

Чен і співавтори[10] Запропонували модель управління доступом на основі можливостей для розподіленого середовища Інтернету речей. Вона підтримує груповий доступ за допомогою одного токена і гарантує повну безпеку за допомогою IPsec. Відправник запиту може використовувати один токен для групового доступу (Група пристроїв, що пропонують загальні послуги) для зв'язку з будь-яким пристроєм в групі. В якості ідентифікатора групи доступу використовується мережевий префікс унікального локального

ідентифікатора (ULA). Кожен пристрій в групі ідентифікується ULA. В маркер групового доступу відправник запиту поміщає свій ULA і мережевий префікс групи доступу. Таким чином, пристрої в групі можуть перевіряти токен за допомогою ULA і префікса в токені. Він також може забезпечувати контроль доступу на основі ULA відправника запиту в токені.

Існуючі стандарти, такі як TLS і PKI, стосуються перших трьох областей безпеки, тобто конфіденційності, цілісності і аутентифікації. Однак контроль доступу вимагає уваги. Оскільки в мультиагентних системах різні агенти мають різні ролі, вони вимагають різних рівнів доступу. Рівера і співавтори [11] запропонували використовувати модель User-Managed Access, яка є профілем OAuth 2.0 і забезпечує різні рівні доступу до різних агентів

OUADDAH і співавтори [12] запропонували нову структуру управління доступом для середовища IoT під назвою "SmartOrBAC", яка заснована на моделі OrBAC. Ця модель використовувала веб-сервіси (підхід RESTFUL) для впровадження політик безпеки.

Організований контроль доступу (OrBAC) має деякі обмеження, такі як, наприклад, він краще працює в централізованій системі, не зачіпає співпрацю між організаціями і подорганізаціямі і не переводить політику безпеки в механізм контролю доступу.

Тому для усунення цих обмежень OrBAC, пропонується SmartOrBAC, який є розширенням OrBAC. SmartOrBAC використовує веб-сервіси для встановлення безпечного співробітництва між різними організаціями. Вони також акцентують увагу на використанні RESTFULL API для обміну інформацією між організаціями, так як він використовує легкий механізм.

Взаємодія між організаціями визначається угодою між ними. Організації спільно визначили правила доступу відповідно до формату OrBAC. У SmartOrBAC контракт не виконується апріорі, але може бути виконаний спонтанно і динамічно. SmartOrBAC забезпечує ефективний контроль доступу для спільних сутностей з низьким енергоспоживанням і обмеженими можливостями енергоспоживання, таких як IoT.

Гіквад та співавтори [13] використовували трирівневу безпечну аутентифікацію Kerberos для системи "розумного будинку" яка використовувала IoT. Вона використовує безпечний хеш-алгоритм SHA 1 і стандарт розширеного шифрування (AES). Однак ні Kerberos не є надійним рішенням для аутентифікації, ні AES не підходить для обмеження IoT-пристроїв.

Перієра та співавтори [14] запропонувала систему контролю доступу для пристроїв з обмеженням доступу на рівні обслуговування. Фреймворк дозволяє здійснювати контроль доступу на кожну послугу з дрібною структурою. Він об'єднує ідеї Kerberos і RADIUS систем контролю доступу для надійної інфраструктури контролю доступу. Він використовує кращі можливості Kerberos, Constrained Application Protocols (CoAP) і RADIUS для створення малопотужної платформи для аспектів контролю доступу та аутентифікації. CoAP клієнт отримує квиток з сервера CoAP і використовує цей квиток в кожному наступному запиті. 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 права, тайм-аут квитка, групу і т.д.

CoAP-NAS висилає користувачеві квиток для подальших запитів. На етапі контролю доступу сервер буде відповідати правильним повідомленням тільки в тому випадку, якщо повідомлення запиту має дійсний квиток, в іншому випадку він видасть повідомлення про помилку.

Легка, безпечна і масштабована схема групової аутентифікації на основі порогової криптографії (TCGA) представлена Махалі та співавторами [15], вона перевіряє ідентичність всіх вузлів груповий комунікації в IoT. Групова аутентифікація знижує накладні витрати на рукостискання, що забезпечує менше використання ресурсів і допомагає економити електроенергію. Ця схема захищена від атаки людини в середній.

Панвар і співавтори [16] запропонували механізм безпеки Інтернету речей з використанням цифрових сертифікатів із захистом на транспортному

рівні (DTLS). Для безпечного зв'язку в IoT аутентифікація здійснюється за допомогою цифрових сертифікатів, наданих центром сертифікації, що робить аутентифікацію більш надійною і замінює собою механізм спільно використовуваних ключів в DTLS. Клієнт / сервер аутентифіковується, перевіряючи підпис наступними кроками. 1: клієнт відправляє запит на сервер. 2: Сервер відправляє свій сертифікат клієнту. 3: Клієнт перевіряє сертифікат, розшифровуючи його відкритим ключем сервера. 4: Після верифікації клієнт відправляє свій власний сертифікат на сервер. 5: сервер перевіряє за допомогою тієї ж процедури, після чого вони можуть почати спілкування.

Сантос та співавтори [17] запропонували схему надання сильної безпеки для системи "розумного будинку". Заропонована система заснована на концепції AllJoyn і використовує криптографію еліптичних кривих для аутентифікації. Система працює в мережі Wi-Fi, в якій є вузол Wi-Fi шлюзу, який відповідає за початкове налаштування системи, аутентифікацію пристроїв IoT і надає користувачеві можливість керувати системою за допомогою мобільного пристрою за допомогою Android застосунку. Процес аутентифікації складається з двох етапів: Мобільний пристрій до IoT пристрою (користувач завантажує ідентифікатор і попередньо виданий ключ і після взаємної аутентифікації домашні облікові дані надаються пристрою IoT) і шлюз до пристрою IoT (пристрій IoT підключається до шлюзу і аутентифікує його за допомогою інформації , що відправляється мобільним пристроєм користувача). Після цього відбувається зашифроване спілкування.

Лі та співавтори [18] запропонували легкий протокол аутентифікації, покращивши оригінальну систему безпеки RFID на базі IoT .У існуючому протоколі RFID аутентифікація здійснюється без шифрування, що є недоліком безпеки. Для вирішення цієї проблеми пропонується легкий криптографічний протокол на основі методу XOR, за яким шифровані паролі використовуються для аутентифікації.

Висновки до розділу 2

В даному розділі проаналізовано основні механізми безпеки IoT, базову архітектуру IoT (трьохрівневу архітектуру) і обговорено проблеми безпеки, атаки і вимоги безпеки на кожному рівні архітектури. Також розглянуто різні схеми аутентифікації присутні в IoT, різні існуючі моделі аутентифікації і контролю доступу.

Результатом аналізу є висновок, що аутентифікація є основним механізмом безпеки, який повинен застосовуватися на різних рівнях архітектури. Через аналіз даних автентифікації можна виявити аномальну активність і, відповідно, вчасно зреагувати на певні типи атак. Своєчасне виявлення аномалій є важливим аспектом ефективного реагування на загрози.

3 АНАЛІЗ ДАНИХ ТА РОЗРОБКА КЛІЄНТСЬКОЇ ПРОГРАМИ З ВИКОРИСТАННЯМ AZURE ДЕТЕКТОРА АНОМАЛІЙ

3.1 Загальні відомості про детектор аномалій

Детектор аномалій дозволяє відстежувати і виявляти відхилення в даних часових рядів за допомогою машинного навчання. Детектор аномалій адаптується шляхом автоматичного визначення і застосування відповідних моделей для заданих даних, незалежно від галузі, сценарію або обсягу даних. Використовуючи дані часових рядів, визначає межі для виявлення аномалій, очікувані значення і точки даних, які є аномаліями.

Детектор аномалій має 4 функції:

1. Виявлення аномалій по мірі їх появи в режимі реального часу.
2. Виявлення аномалій набору даних у пакетному режимі.
3. Отримання додаткових відомостей про дані.
4. Налаштування меж виявлення аномалій.

Виявлення аномалій на основі щільності базується на алгоритмі k-найближчих сусідів.

Припущення: Звичайні точки збору даних розташовуються навколо щільного району, а аномалії - далеко.

Найближчий набір точок даних оцінюється з використанням балів, які можуть бути Евклідовою відстанню або схожою в залежності від типу даних (категорії або числа). Їх можна розділити на два алгоритми:

1. K-найближчих сусідів: k-NN - це простий, непараметричний метод ледачого навчання. Сусіди утворюються із множини об’єктів, класи яких уже відомі, і, виходячи із заданого значення k ($k \geq 1$), визначається який із класів найбільш багатозначний. Якщо $k = 1$, то об’єкт відноситься до класу одного найближчого сусіда. Цей алгоритм є одним із найпопулярніших і водночас простим серед інших

алгоритмів класифікації, часто саме він є першим при ознайомленні з машинним навчанням. Але також часто цей алгоритм допомагає в вирішенні масштабних задач і входить в арсенал досвідчених дослідників в даній області. Використовується для класифікації даних на основі подібності в дистанційних метриках, таких як Евклідова, Манхеттова, Мінковська або Хаммінгова відстань.

2. Відносна щільність даних: Більш відома як місцевий фактор викиду (LOF). Ця концепція заснована на метриці відстані, званої відстанню досяжності.

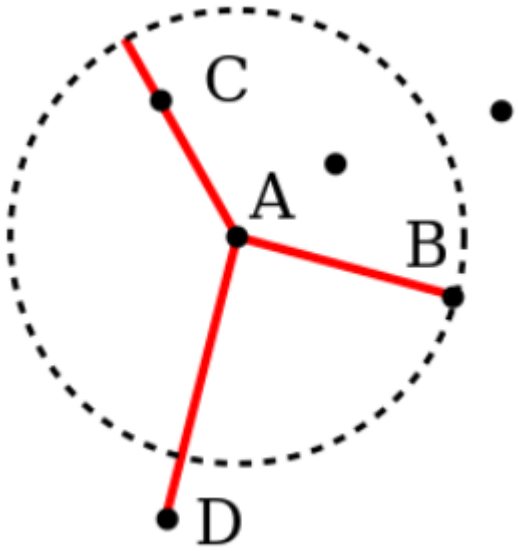

Рисунок 3.1 – Відстань досягаємості. Об'єкти B і C мають однакову відстань досяжності(k=3) в той час як D не є k-найближчим сусідом

Формальний опис(рисунок 3.1): Нехай k-distance(A) буде відстань від об'єкта A до k-го найближчого сусіда. В набір k найближчих сусідів входять всі об'єкти на цій відстані, які можуть в разі "вузла" містити більше k об'єктів. Позначимо множину k найближчих сусідів як N_k (A). Ця відстань використовується для визначення відстані досяжності(англ. Reachability-distance):

Reachability-distance$_k$ (A, B) = max{k-distance(B), d(A, B)}

Відстань досяжності об'єкта A з B є істинною відстанню двох об'єктів. Об'єкти, які належать до k найближчих сусідів точки B, вважаються такими, що знаходяться на однаковій відстані для отримання стабільніших результатів. Зауважимо, що ця відстань не є відстанню в математичному сенсі, оскільки вона не симетрична. Локальна щільність досяжності об'єкта A визначається як

$$\operatorname{lrd}_k(A) := 1/\left(\frac{\sum_{B \in N_k(A)} \text{reachability-distance}_k(A,B)}{|N_k(A)|}\right) \tag{3.1}$$

яка є зворотним значенням середньої відстані досяжності об'єкта A від його сусідів. Зауважимо, що це не є середньою відстанню досяжності сусідів з точки A, а є відстанню, на якій A може бути «досягнуто» з його сусідів. З дублікатами точок це значення може стати нескінченним.

Локальні щільності досяжності потім порівнюються з локальними щільностями досяжності сусідів

$$\operatorname{LOF}_k(A) := \frac{\sum_{B \in N_k(A)} \frac{\operatorname{lrd}(B)}{\operatorname{lrd}(A)}}{|N_k(A)|} = \frac{\sum_{B \in N_k(A)} \operatorname{lrd}(B)}{|N_k(A)|} / \operatorname{lrd}(A) \tag{3.2}$$

середня локальна щільність досяжності сусідів, поділена на локальну щільність досяжності самого об'єкта. Значення, приблизно рівне 1, означає, що об'єкт можна порівняти з його сусідами (і тоді він не є викидом). Значення менше 1 означає щільну область (яка може бути серцевиною), а значення, істотно більші 1, свідчать про викиди.

Кластеризація є однією з найпопулярніших концепцій в області неконтрольованого навчання.

Припущення: Аналогічні точки даних, як правило, належать до аналогічних груп або кластерів, що визначаються їх віддаленістю від місцевих Центроїд.

K-засіб - це широко використовуваний алгоритм кластеризації. Він створює 'k' схожі кластери точок даних. Випадки, що виходять за рамки цих груп, можуть бути потенційно відзначені як аномалії.

Машина з вектором підтримки - ще один ефективний метод виявлення аномалій. SVM зазвичай асоціюється з контрольованим навчанням, але існують розширення (наприклад, OneClassCVM), які можуть бути використані для виявлення аномалій як неконтрольованих проблем (в яких дані навчання не позначені). Алгоритм вивчає межу для угруповання звичайних екземплярів даних за допомогою навчального набору, а потім, використовуючи тестовий екземпляр, сам налаштовується на виявлення аномалій, які виходять за межі досліджуваного регіону.

Залежно від конкретного випадку використання, виходом детектора аномалій можуть бути числові скалярні значення для фільтрації по порогам домену або текстовим міткам (наприклад, двійкові / мультиметрові мітки).

3.2 Виявлення аномалій в даних часових рядів з використанням REST API Детектора аномалій

Для будь-якої системи виявлення аномалій часових рядів, що працює в великомасштабному виробництві, існує досить багато проблем, особливо в наступних трьох областях:

1. Відсутність міток – При генеруванні кожної секунди клієнтами, службами та датчиками сигналів, їх величезний обсяг унеможливлює нанесення міток вручну.

2. Узагальнення - Маючи справу з реальними даними, існує багато різних типів часових рядів з різними характеристиками, що ускладнює узагальнення і пошук правильного рішення для всіх проблем.

3. Ефективність - Для будь-якої системи виявлення аномалій ефективність є одним з ключових завдань. Очікується, що система буде мати низьку обчислювальну вартість і низьку затримку в обслуговуванні.

З огляду на те, що галузь Інтернету речей розвивається і кількість систем збільшується, ефект атак може бути не тільки дуже потужним і впливовим, але і дуже масштабним. У зв'язку з цією проблемою пропонується метод забезпечення безпеки мережі Інтернету речей, який є швидким в плані реагування і простим для впровадження.

Запропонований метод працює наступним чином(рисунок 3.2):

1. Збираються IPsec заголовки автентифікації
2. На основі цих заголовків будуються часові ряди які містять дані про кількість спроб автентифікації за певний період часу
3. Дані часових рядів передаються в детектор аномалій, де вони аналізуються, виявляються аномалії. Інформація про час та значення виявлених аномалій є вихідними даними.
4. На основі отриманих результатів робиться висновок про поломку пристрою, або ймовірну атаку

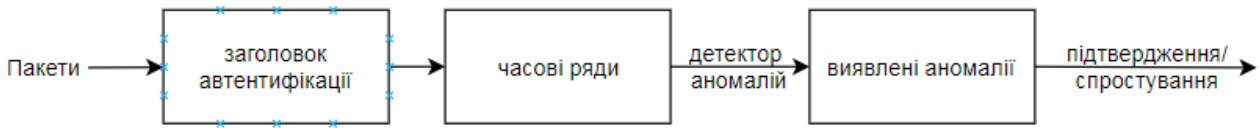

Рисунок 3.2 – Блок-схема алгоритму виявлення аномалій за допомогою детектору аномалій в мережі Інтернету речей

Оскільки автентифікація присутня на всіх рівнях архітектури Інтернету речей, то аналіз заголовків автентифікації протоколу IPsec є ефективним шляхом взаємодії з мережею.

Для виявлення відхилень запропоновано використовувати один з сервісів Microsoft Azure, а саме детектор аномалій.

3.2.1 Методика підключення детектору аномалій

Реєстрація в Azure Cognitive Services та створення ресурсу Детектора аномалій на порталі Azure.

Для підключення застосунку до API “Детектор аномалій” необхідно ключ і кінцева точка із створеного ресурсу(рисунок 3.1).

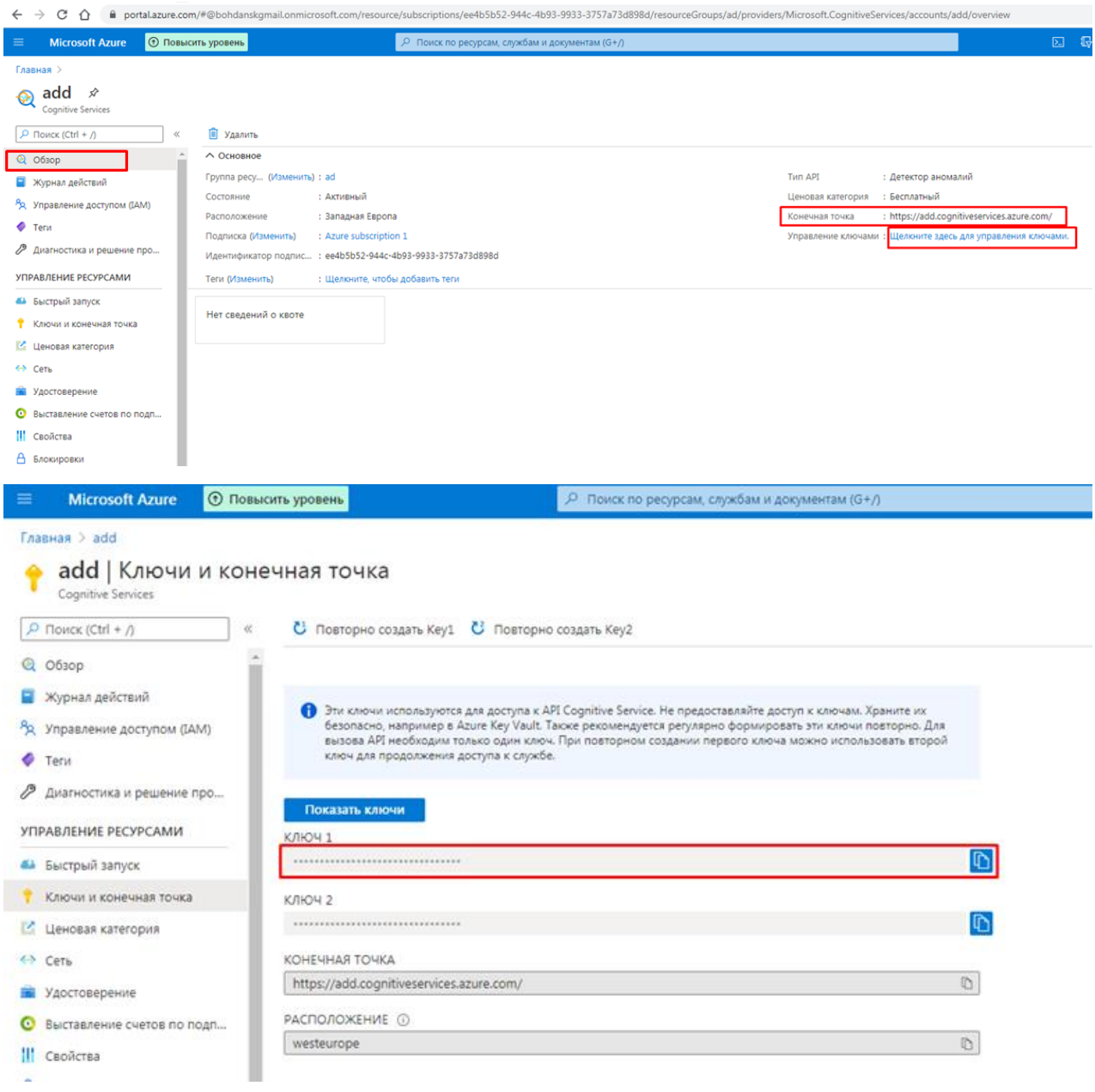

Рисунок 3.1 – ключ та кінцева точка детектору аномалій

Також попередньо необхідно встановити Python та бібліотеку запитів(requests) для Python.

Дані часових рядів передаються у форматі JSON-файлу.

Використовуючи ключ та кінцеву точку із створеного ресурсу необхідно створити дві змінні середовища для перевірки достовірності(рисунок 3.2).

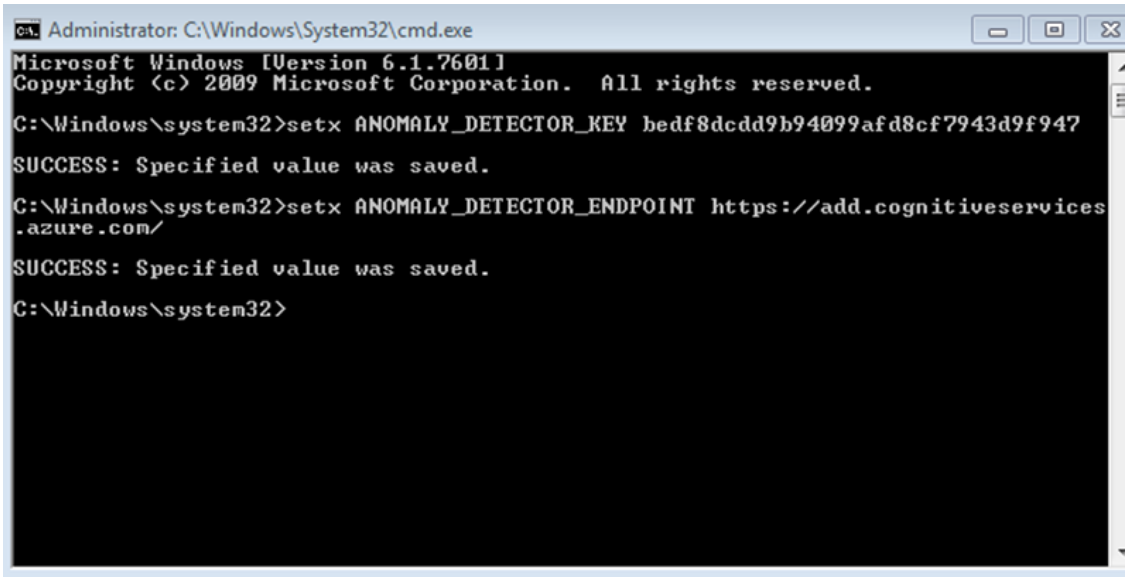

Рисунок 3.2 – Anomaly_detector_key – ключ ресурсу для перевірки достовірності; Anomaly_detector_endpoint – кінцева точка ресурсу для відправлення запитів API.

3.2.2 Розробка клієнтської програми для передачі датасетів до детектору аномалій

Налаштування та передача датасету у вказаний сервіс відбувається за допомогою наступного коду на мові Python:

```
import os
import requests
import json
batch_detection_url = "/anomalydetector/v1.0/timeseries/entire/detect"
latest_point_detection_url = "/anomalydetector/v1.0/timeseries/last/detect"
```

створили змінну середовища для ключа та кінцевої точки

```
endpoint = "https://anomalydetector12.cognitiveservices.azure.com/"
subscription_key = "24de67bd5cd143ffbc311be7ab726ddf"
```

шлях до даних часових рядів у форматі JSON

```
data_location = "request-data.json"
file_handler = open(data_location)
json_data = json.load(file_handler)
```

Функція для відправлення запитів send_request та функція виявлення аномалій та запису результату до файлу detect_batch, функція перевірки останньої точки на наявність аномалії detect_latest

```
def send_request(endpoint, url, subscription_key, request_data):
    headers = {'Content-Type': 'application/json', 'Ocp-Apim-Subscription-Key': subscription_key}
    response = requests.post(endpoint+url, data=json.dumps(request_data), headers=headers)
    return json.loads(response.content.decode("utf-8"))
def detect_batch(request_data):
    f = open('text.txt', 'w')
    f.write("Detecting anomalies as a batch")
    result = send_request(endpoint, batch_detection_url, subscription_key, request_data)
    f.write(json.dumps(result, indent=4))
    if result.get('code') is not None:
        f.write("Detection failed. ErrorCode:{}, ErrorMessage:{}".format(result['code'], result['message']))
```

```
    else:
        # Знаходимо і відображаємо позицію аномалії в датасеті
        anomalies = result["isAnomaly"]
        f.write("Anomalies detected in the following data positions:")
        for x in range(len(anomalies)):
            if anomalies[x]:
                f.write(str(x) + "," + str(request_data['series'][x]['value']) + "\n")
                # f.write(request_data['series'][x]['value'])
    f.close()
def detect_latest(request_data):
    f = open('text.txt', 'a')
    f.write("Determining if latest data point is an anomaly")
    result = send_request(endpoint, latest_point_detection_url, subscription_key,
request_data)
    f.write(json.dumps(result, indent=4))
    f.close()
```

Виявлення аномалійшляхом виклику попередньо створених функцій

```
detect_batch(json_data)
detect_latest(json_data)
```

За допомогою описаних функцій файл датасету передається до детектору аномалій в якому відбувається виявлення відхилень і на виході отримуємо інформацію про дані які відмінні від норми(рисунок 3.3).

```
}Anomalies detected in the following data positions:3,25948736
18,33631649
21,38144434
22,34662949
23,24623684
24,26530491
25,35445003
28,30744783
29,25825128
30,21244209
31,22576956
32,31957221
35,32383350
44,22504059
```

Рисунок 3.3 – позиції виявлених аномалій в датасеті

3.2.3 Розробка програми для візуалізації результату на основі JupyterLab

Для побудови графіків необхідно встановити anaconda navigator через який отримаємо доступ до JupyterLab(рисунок 3.4).

JupyterLab - це інтерактивне середовище розробки для роботи з блокнотами, кодом і даними. Найголовніше, що JupyterLab має повну підтримку блокнотів Jupyter. Крім того, JupyterLab дозволяє використовувати текстові редактори, термінали, засоби перегляду файлів даних і інші компоненти поряд з блокнотами в розбитому на вкладки робочому середовищі.

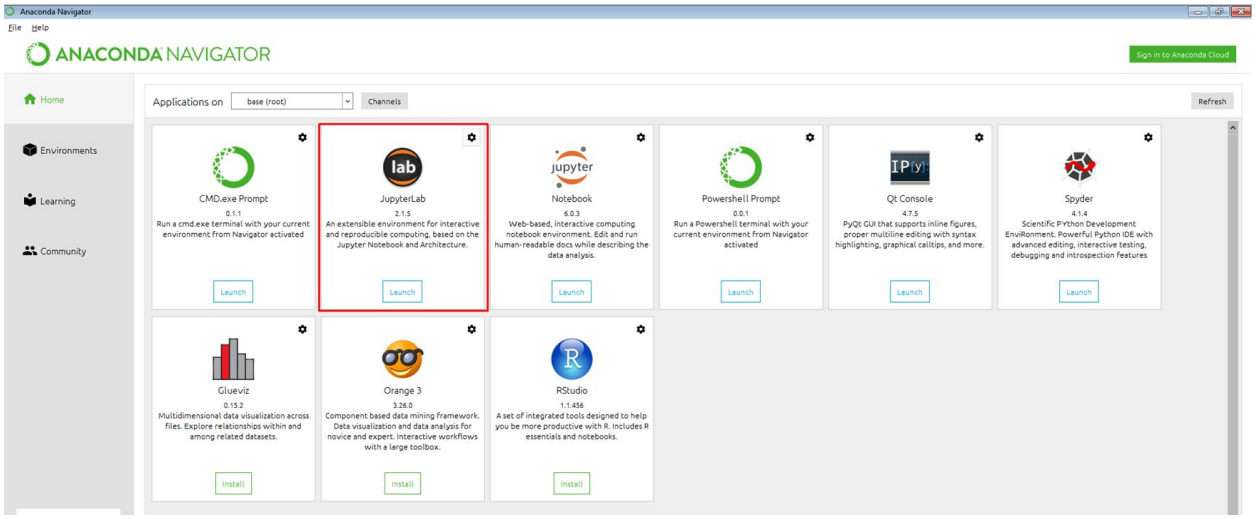

Рисунок 3.4 – запуск jupyterLab через інтерфейс anaconda navigator

Функція для побудови графіку:

```
def build_figure(sample_data, sensitivity):
    sample_data['sensitivity'] = sensitivity
    result = detect(endpoint, subscription_key, sample_data)
    columns = {'expectedValues': result['expectedValues'], 'isAnomaly':
result['isAnomaly'], 'isNegativeAnomaly': result['isNegativeAnomaly'],
        'isPositiveAnomaly': result['isPositiveAnomaly'], 'upperMargins':
result['upperMargins'], 'lowerMargins': result['lowerMargins'],
```

```
        'timestamp': [parser.parse(x['timestamp']) for x in sample_data['series']],
        'value': [x['value'] for x in sample_data['series']]}
    response = pd.DataFrame(data=columns)
    values = response['value']
    label = response['timestamp']
    anomalies = []
    anomaly_labels = []
    index = 0
    anomaly_indexes = []
    p = figure(x_axis_type='datetime', title="Batch Anomaly Detection ({0} Sensitvity)".format(sensitivity), width=800, height=600)
    for anom in response['isAnomaly']:
      if anom == True and (values[index] > response.iloc[index]['expectedValues'] + response.iloc[index]['upperMargins'] or
                  values[index] < response.iloc[index]['expectedValues'] - response.iloc[index]['lowerMargins']):
          anomalies.append(values[index])
          anomaly_labels.append(label[index])
          anomaly_indexes.append(index)
      index = index+1
    upperband = response['expectedValues'] + response['upperMargins']
    lowerband = response['expectedValues'] -response['lowerMargins']
    band_x = np.append(label, label[::-1])
    band_y = np.append(lowerband, upperband[::-1])
    boundary = p.patch(band_x, band_y, color=Blues4[2], fill_alpha=0.5, line_width=1, legend='Boundary')
    p.line(label, values, legend='Value', color="#2222aa", line_width=1)
    p.line(label, response['expectedValues'], legend='ExpectedValue', line_width=1, line_dash="dotdash", line_color='olivedrab')
    anom_source = ColumnDataSource(dict(x=anomaly_labels, y=anomalies))
```

```
anoms = p.circle('x', 'y', size=5, color='tomato', source=anom_source)
p.legend.border_line_width = 1
p.legend.background_fill_alpha  = 0.1
show(p, notebook_handle=True)
```

Викликаємо вказану функцію передаючи файл датасету та задаючи чутливість отримуємо візуалізацію результатів для погодинної вибірки

```
sample_data = json.load(open('sample_hourly.json'))
sample_data['granularity'] = 'hourly'
sample_data['period'] = 24
# 95 sensitivity
build_figure(sample_data,95)
```

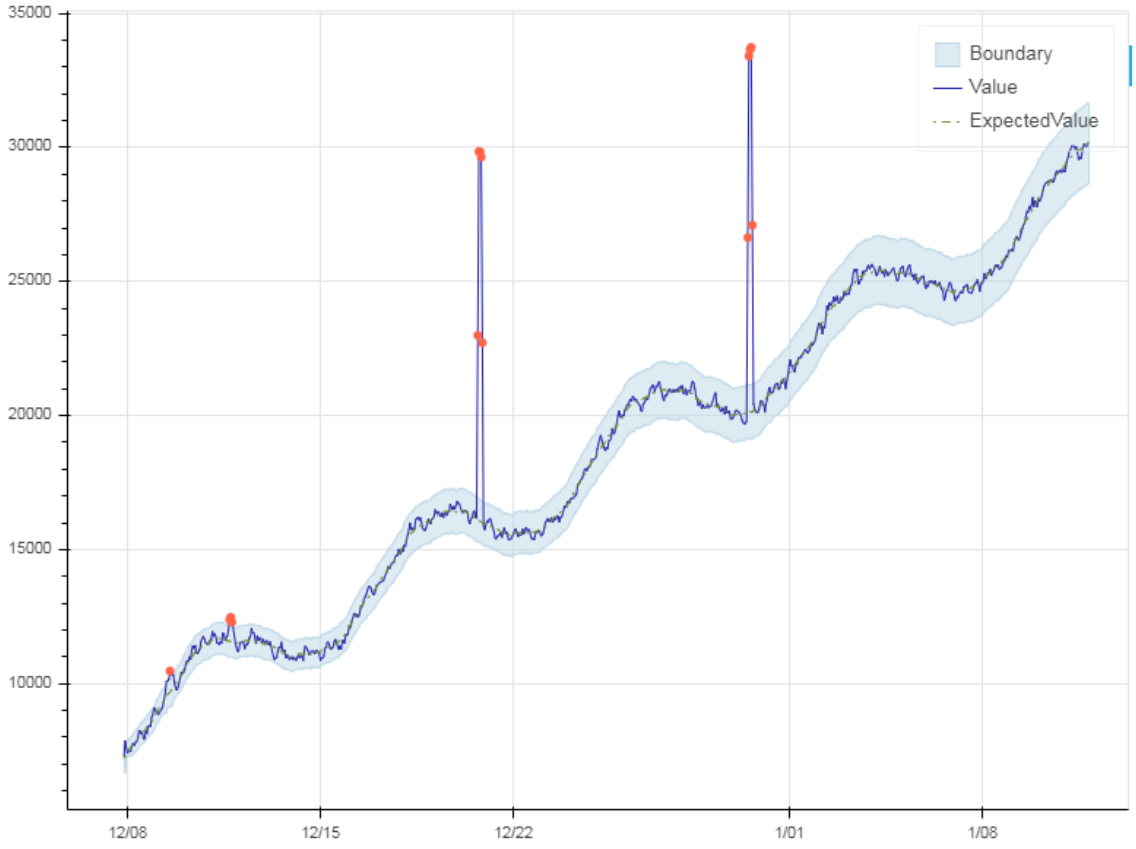

Рисунок 3.2 – часовий ряд з погодинною частотою вибірки(чутливість 95)

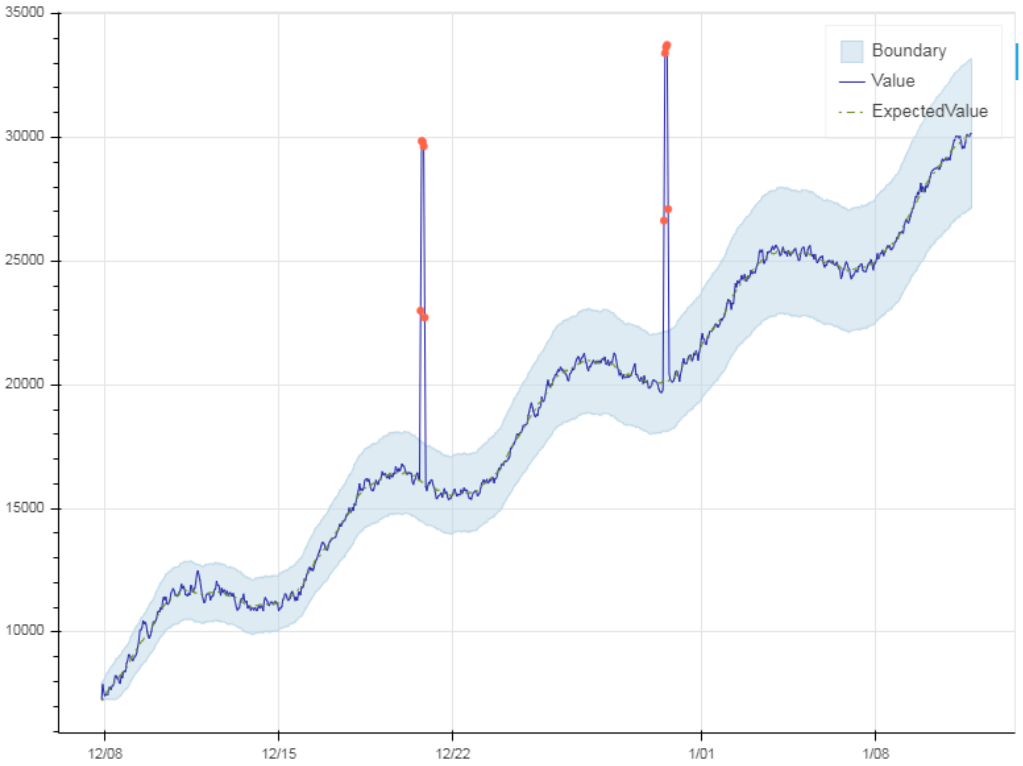

Рисунок 3.3 – часовий ряд з погодинною частотою вибірки(чутливість 90)

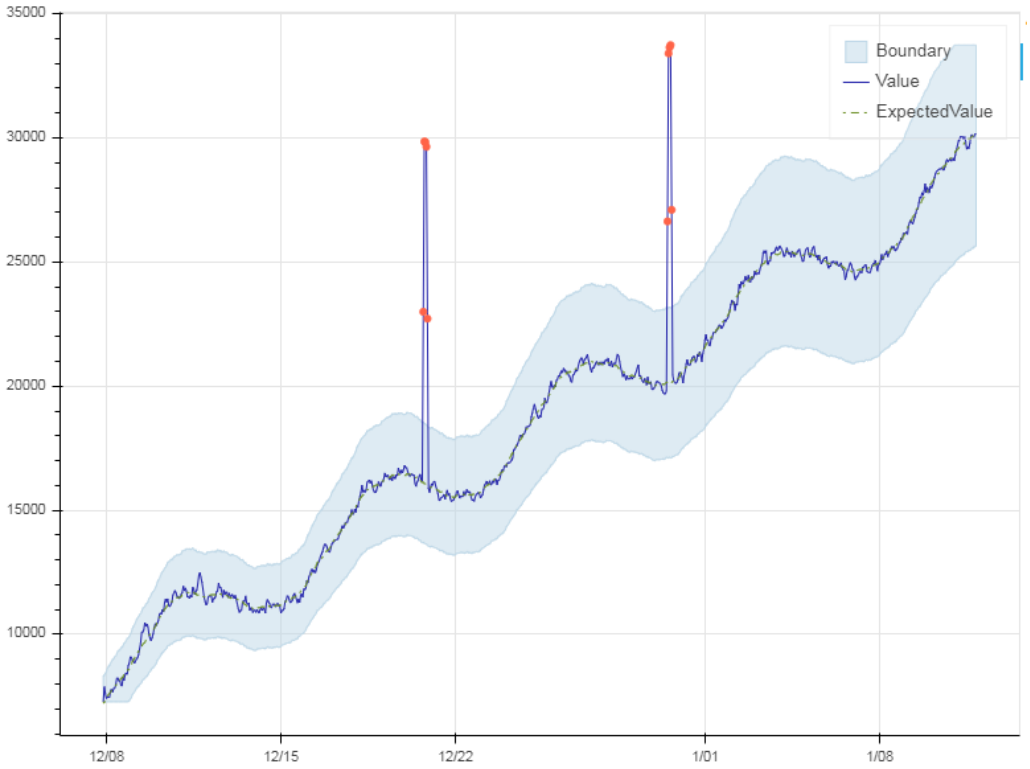

Рисунок 3.4 – часовий ряд з погодинною частотою вибірки(чутливість 85)

Викликаємо вказану функцію передаючи файл датасету та задаючи чутливість отримуємо візуалізацію результатів для поденною вибірки

```
sample_data = json.load(open('sample.json'))
sample_data['granularity'] = 'daily'
# 95 sensitivity
build_figure(sample_data,95)
```

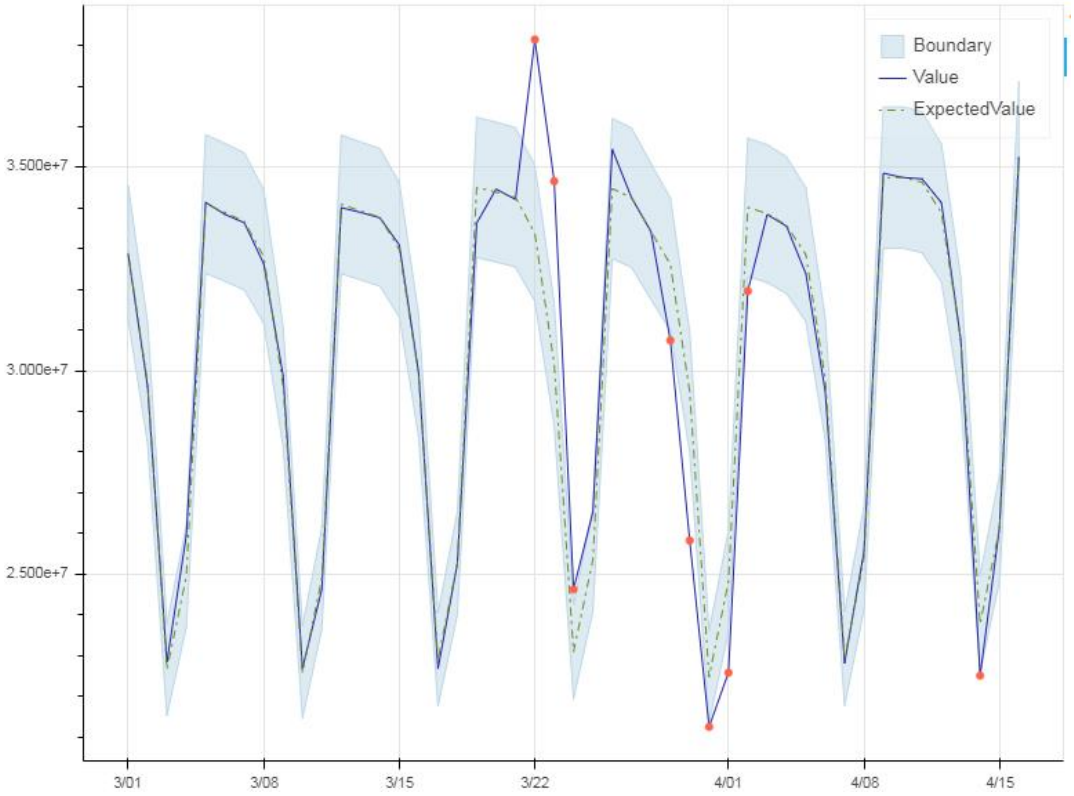

Рисунок 3.5 – часовий ряд із поденною частотою вибірки(чутливість 95)

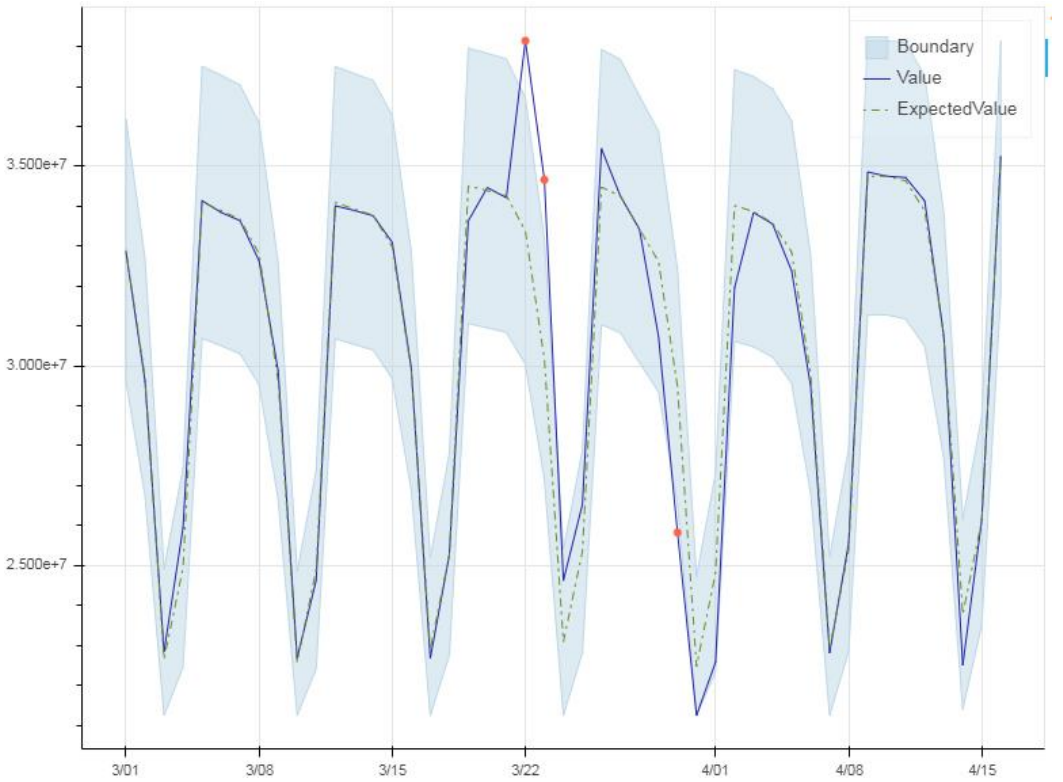

Рисунок 3.6 – часовий ряд із поденною частотою вибірки(чутливість 90)

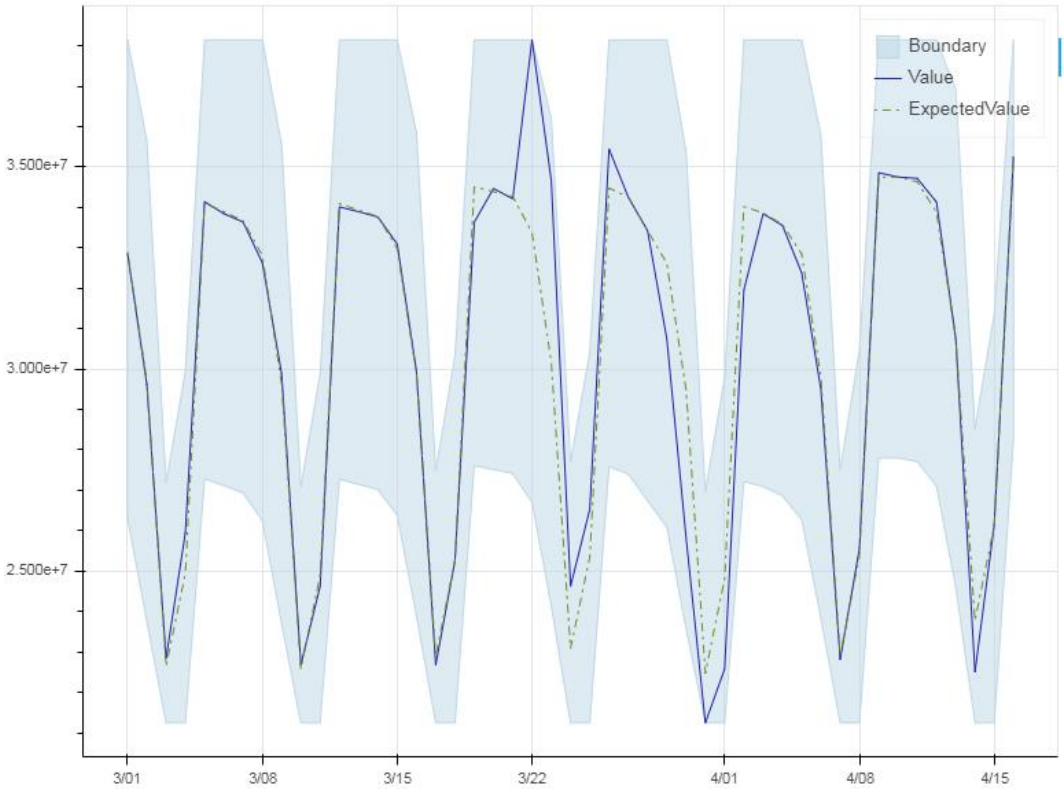

Рисунок 3.7 – часовий ряд із поденною частотою вибірки(чутливість 85)

На рисунках 3.2 – 3.7 відображено графіки даних з датасету.

Таким чином виявлено аномалії в даних автентифікації. Провівши подальший аналіз отриманих даних можна зробити висновки про спробу атаки, взлому чи інших небажаних дій.

Висновки до розділу 3

Багато мереж характеризуються чітко визначеними шаблонами поведінки, а відхилення в таких системах легко ідентифікуються. Широке різноманітність промислових протоколів і протоколів IoT ускладнює ситуацію, але нові технічні рішення, які використовують просунуте машинне навчання, успішно вирішують аналітичні завдання.

В результаті роботи проаналізовано аспекти безпеки пристроїв інтернету речей та розроблено метод аналізу даних для пошуку аномалій в даних автентифікації. Запропоноване рішення є засобом захисту в пасивному режимі «виявлення», такий підхід веде себе не так агресивно, ніж в активному режимі «запобігання», оскільки помилкові спрацьовування не впливатимуть на роботу системи загалом. Розроблена програма апробована на реальних даних

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Впровадження методу для фільтрації даних в сферу IoT з метою підвищення її ступеню захищеності, шляхом виявлення атак за допомогою аудиту аутентифікації	1. Підвищення захисту інформації компаніями великого та середнього розмірів	Швидке виявлення ймовірних проблем безпеки інформаційних мереж
	2. Окремі користувачі для власних потреб	Виявлення підозрілої активності та підвищення рівня безпеки даних

Таблиця 4.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W (слабка сторона)	S (сильна сторона)
		Мій проект	AWS IoT Device Defender		
1.	Економічні	Витрати на купівлю ліцензій, серверів для обробки даних, витрати на	витрати на персонал, обслуговування серверів	Необхідне збільшення витрат на сервери при зростанні кількості	Можливість масштабованості, експоненційний ріст

		персонал 20000$		користувачів	користувачів, вигідність рішення
2.	Технічні	Використання хмарного сервісу для пошуку аномалій	Використання конфігураційних файлів для перевірки стану компонентів системи	Необхідність взаємодії з стороннім сервісом	Обширна аудиторія користувачів, легка масштабованість
3.	надійність	Забезпечення захищеності інформації при її обробці та передачі	Робота виключно з кофігураційними файлами	Залежить від доступності серверів та сервісу	Високий рівень захищеності інформації за рахунок використання хмарних технологій
4	безпеки	Задіяння хмарного сервісу, зосередження на атаках аутентифікац	Зосередження на стані конфігураційних файлів компонентів системи	Витрати на використання хмарних сервісів	Використання хмарних сервісів підвищує швидкіст

		її			ь виявленн я загроз

4.2 Технологічний аудит ідеї проекту

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Налаштування системи аналізу даних	Конфігураційний файл на мові Python	Документація платформи	доступна
2	Збір даних що будуть аналізуватися	Протоколи аутентифікації	Використання відомих алгоритмів	доступна
3	Використання хмарних технологій	Технологія Azure Anomaly Detector	Оформлення необхідних сервісів і їх налаштування	доступна
Обрана технологія реалізації ідеї проекту: Необхідні технології наявні та доступні, реалізація полягає в об’єднанні вказаних засобів і інструментів.				

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 4.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика

1	Кількість головних гравців, од	4 од
2	Загальний обсяг продаж, грн/ум.од	2 млн
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	немає
5	Специфічні вимоги до стандартизації та сертифікації	Надання запропонованому рішенню захисту відповідних сертифікатів
6	Середня норма рентабельності в галузі (або по ринку), %	60%

Середня норма рентабельності в галузі (або по ринку) порівнюється із банківським відсотком на вкладення. За умови, що останній є вищим, можливо, має сенс вкласти кошти в інший проект.

За результатами аналізу таблиці можна стверджувати про привабливість проекту для виходу на ринок за попереднім оцінюванням.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Підвищення рівня безпеки системи IoT шляхом виявлення аномалій	Будь-яка система IoT, від приватного використання до підприємств і корпорацій	Цільовою групою є корпорації які прагнуть зменшити втрати від атак на системи IoT	- Достатньо велича точність та швидкість виявлення аномалій

			шляхом їх своєчасного виявлення. Другою групою є звичайні користувачі наприклад розумного будинку, які прагнуть забезпечити себе від небажаного втручання в системи їх оточення	

Таблиця 4.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Недовіра з боку користувача до систем виявлення аномалій	Зміна кількості користувачів в нижчу сторону, викликана порушеннями конфіденційності інформації	При відсутності реакції на недоліки існуючих рішень, може виникнути потреба для створення власного детектору аномалій

Таблиця 4.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії

1	Збільшення частоти несанкціонованих втручань у систему	Збільшення потреб користувача у виявленні несанкціонованих втручань	Маркетинг, збільшення бази клієнтів
2	Збільшення обрахункових потужностей пристроїв системи	Сприяння виробників пристроїв розробці та впровадженню нових засобів та рішень	Розширення розмірів розвитку

Таблиця4.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - олігополія	Наявність на ринку декількох конкурентів	Ймовірність зайняти позицію у сфері за рахунок новизни рішення
2. За рівнем конкурентної боротьби - міжнародний	Універсальність надання послуг за кордоном	Можливість суттєвого розширення ринку збуту послуги
3. За галузевою ознакою - внутрішньогалузева	Послуга є застосовувана в сфері IoT	Наявність широкого ринку збуду послуги
4. Конкуренція за видами товарів: - товарно-видова	Послуга використовується для зодоволення потреб клієнтів, але має відмінності від рішень	Залучення рішення до малих систем IoT

	конкурентів на корись якості існуючих сервісів	
5. За характером конкурентних переваг - нецінова	Ціна не основний фактор для клієнта	Збільшення показників якості продукту та його функціоналу збільшює базу клієнтів
6. За інтенсивністю - марочна	Визначення підходів для надання сервісу	Збільшення якості послуг, які надаються

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Навести перелік прямих конкурентів: Відсутні	Визначити бар'єри входження в ринок: AWS IoT Device Defender	Визначити фактори сили постачальників: не впливають	Визначити фактори сили споживачів: Визнання продукту, або відмова від його використання	Фактори загроз з боку замінників: відсутні
Висновки:	Визначити інтенсивність конкурентної	При наданні якісних послуг можливість	Не диктують	Вимоги до якості продукту, точності	Наявність вбудованих рішень в системи

	боротьби з боку прямих конкурентів: Відсутня	виходу на ринок значна		його роботи	IoT

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Доступність	Продукт легкий в використанні, що сприяє збільшенню користувачів
2	Відсутність прямих конкурентів	Унікальність розробленого рішення
3	Гнучкість підходу до розробки послуг	Використовуваний підхід забезпечує максимальну гнучкість рішень та підлаштування під унікального клієнта

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим рішенням						
			–3	–2	–1	0	+1	+2	+3
1	Ціна та змінні витрати	17			+				
2	Запропонований підхід до послуг	11		+					
3	Доступність ресурсів для якісних розробок послуг	5						+	
4	Доступ до ресурсів у конкурентів	15							+

Таблиця 4.12. SWOT- аналіз стартап-проекту

Сильні сторони: Легкість роботи з запропонованою системою, зрозумілий інтерфейс	Слабкі сторони: Низька популярність, потрібно напрацьовувати репутацію
Можливості: розвиток безпеки у сфері IoT	Загрози: використання користувачами безкоштовних аналогів

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Побудова рішення використовуючи компоненти одного постачальника	Висока ймовірність отримання ресурсів та залучення підтримкою вендора	7 місяців
2	Побудова гібридного сервісу	Низька ймовірність отримання ресурсів	4 роки
3	Розробка власного сервісу на основі існуючого рішення та залучення клієнтів	Висока ймовірність отримання ресурсів при умові успішного впровадження попередніх рішень	6 років

З означених альтернатив, найкращим рішенням буде початок роботи з використання компонентів одного постачальника з подальшим розширенням і переходом до гібридного сервісу і вподальшому власного сервісу.

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
	Звичайні користувачі	Користувачі зацікавлені у безпеці свого середовища готові до використання продукту	Середній попит у користувачі в невеликих систем без важливих або цінних подій	Конкуренція знаходиться на низькому рівні через низьку ймовірність самостійного виявлення дефектів	Можливі складнощі через відсутність зарекомендованого результату на ринку
	Корпорації та компанії що працюють з системами IoT	Компанії та корпорації що турбуються про безпеку своїх систем, або систем які вони надають	Високий рівень попиту у з’язку з наявністю важливої інформації в системі	Високий рівень конкуренції через можливість компаній самостійного виявлення атак	Ймовірність складнощів пов’язаних зі зміною політики компаній щодо впровадження продукту
Які цільові групи обрано: звичайні користувачі, корпорації та компанії					

представники послуг IoT

Таблиця 4.15. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Побудова рішення використовуючи компоненти одного постачальника	Диференційованого маркетингу	Підвищення рівня виявлень атак	Інструментом реалізації стратегії диференціації є ринкове позиціонування.
2	Побудова гібридного сервісу	Диференційованого маркетингу	Розширення ринку продукту	Стратегія диференціації
3	Розробка власного сервісу на основі існуючого	Диференційованого маркетингу	Виділення власного продукту з усіма перевагами у використанні	Стратегія спеціалізації

	рішення та залучення клієнтів			

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	так	Рішення буде доступне усім користувачам існуєчих систем	ні	Стратегія лідера

Таблиця 4.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Забезпечення додаткового рівня виявлення атак. Відсіюванн	Стратегія диференціації	Надання додаткового рівня виявлення атак	Моніторинг безпеки IoT систем клієнтів Розслідування інцидентів, аудит в сфері інформаційної безпеки

	я хибних спрацювань			

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
	Виявлення атак	Впровадження додаткового рівня безпеки для виявлення атак та своєчасного реагування шляхом аудиту даних аутентифікації	Використання хмарного сервісу з машинним навчанням

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Моніторинг IoT систем клієнтів. Розробка та розгортання хмарної консолі керування безпекою. Можливість розслідування інцидентів, проведення аудитів в сфері інформаційної безпеки		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Використання хмарних сервісів для розміщення рішення	М	Вр/Тх/Тл
	2. Швидкість роботи	М	Тх/Е/Ор

	рішення для користувача		
	3. Надання доступу до хмарної консолі з будь=якої точки світу	М	Тл/Е
	4. Аутентифікація пристроїв	М	Тх/Е
ІІІ. Товар із підкріпленням	До продажу : використання безкоштовної версії з тимчасовим доступом, з обмеженим функціоналом		
	Після продажу: надання послуг без обмежень, проведення додаткового маркетингу		
За рахунок чого потенційний товар буде захищено від копіювання:захист інтелектуальної власності, унікальний підхід до виявлення атак			

Таблиця 4.20. Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Відсутній	1000 ум.од.	Високий або середній	Ціна підписки на сервіс моніторингу 300-750 ум.од.

Таблиця 4.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Прийняття рішення про необхідність системи моніторингу	Пошук перспективних засобів просування товарів; вибір	Довірені канали збуту	Залучення компаній посередників та партнерів

		посередників; розробка та вдосконалення політики маркетингу		для формування системи збуту

Таблиця 4.22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Дослідження властивостей та якості рішення, можливість тестування продукту, прийняття рішення про необхідність використання продукту для задоволення своїх потреб	Системи IoT	Оцінювання ефективності контролю безпеки, зменшення впливу атак у системах IoT	Донести переваги до потенційних користувачів	Демонстрація актуальних атак та їх виявлення за допомогою нашого методу

Висновки до розділу 4

В рамках розділу проведено аналіз та розробку бізнес-проекту, слід відзначити що проект має непогані можливості виходу на ринок. Також присутня перспектива впровадження з огляду на потенційні групи клієнтів. Проблема є актуальною оскільки кількість користувачів IoT системами зростає, тому системи захисту будуть користуватися попитом.

Проект має гарну рентабельність на ринку послуг. Також присутні негативні аспекти пов’язані з конкурентами. Для впровадження рішення краще почати з інтеграції в невеликі системи IoT для рекомендації себе та набуття популярності, з подальшим виділенням рішення в окрему систему. Подальша імплементація є доцільною. Аналіз стартап-проекту показує актуальність та можливість реалізації даного проекту.

ВИСНОВКИ

В роботі було досліджено та проаналізовано архітектуру Інтернету речей, можливі загрози і атаки на його систему, вимоги безпеки на кожному рівні архітектури, стан захищеності протоколів IoT. Виявлено що аутентифікація є основним механізмом безпеки, який повинен застосовуватися на різних рівнях архітектури і, відповідно, через автентифікацію можна виявити і вчасно зреагувати на певні типи атак.

Запропоновано рішення є засобом захисту в пасивному режимі «виявлення», такий підхід веде себе не так агресивно, ніж в активному режимі «запобігання», оскільки помилкові спрацьовування не впливатимуть на роботу системи загалом.

Реалізовано запропонований метод та підготовлено результати роботи даного методу. За результатами реалізації зроблено висновок, що метод є працездатним, дозволяє ефективно та своєчасно виявити загрози безпеки в мережі Інтернету речей, задовольняє потреби більшості IoT систем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 Загальні поняття Інтернету речей[Електронний ресурс]. - Режим доступу: https://academicfox.com/lektsiya-1-zahalni-ponyattya-internetu-rechej/

2 Росляков А. В. ИНТЕРНЕТ ВЕЩЕЙ : Учебное пособие[Текст]. / А. В. Росляков, С. В. Ваняшин, А. Ю. Гребешков. - Самара : ПГУТИ, 2015 - 136 с.

3 D. Evans, The Internet of things: How the next evolution of the Internet is changing everything [Text]. CISCO, San Jose,CA, USA,White Paper, 2011.

4 L. Atzori, A. Iera, and G. Morabito, The Internet of Things: A survey[Text]. Comput. Netw., vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

5 Гиббс, М. Интернет вещей – не только для «умных» [Текст]. / М. Гиббс // Сети/network world. – 2013. – №3.

6 П'ять основних тенденцій розвитку Інтернету речей в 2018 році[Електронний ресурс]. – Режим доступу https://beasthackerz.ru/uk/kompyuter/internet-veshchei-v-monitoringe-ochen-pechalnyi-primer-nedoocenki.html

7 А. Найдич Большие данные: насколько они большие?[Електронний ресурс]. - Режим доступу: http://compress.ru/article.aspx?id=23469

8 Лекція з курсу Проектування Інтернет речей (IoT)[Електронний ресурс]. Sep 10. 2017 – Режим доступу https://www.slideshare.net/ssuserf405bc/iot-79608563

9 Federico Giannoni, Marco Mancini, Federico Marinelli[Electronic resource]. Nov 30 2018 - Access mode: https://arxiv.org/abs/1812.00890

10 B. Chen, Y L. Huang, M G. Unes, “S-CBAC: A secure access control model for supporting group access for internet of things.”[Text]. 2015 IEEE.

11 D. Rivera, L. Paris, G. Civera, E. Hoz, I. Maestre, “Applying an unified access control for IoT based Intelligent agent system.”[Text]. IEEE international conference on service-oriented computing and application. 2015

12 A. OUADDAH, I. PASQUIER, A. ELKALAM, A. OUAHMAN, Security analysis and proposal of new access control model in the Internet of things[Text]. // 1st International conference on Electrical and Information Technologies ICEIT, 2015.

13 P. Gaikwad, J. Gabhane, S. Golait, 3-level secure Kerberos authentication for smart home system using IoT[Text]. // International conference on next generation computing technologies 2015 (NGCT2015).

14 P. Periera, J. Eliasson, J. Delsing, An authentication and access control framework for CoAP based internet of things[Text]. // Proceedings, IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society

15 P. Mahalle, N. Prasad, R. Prasad, threshold cryptography-based group authentication (TCGA) scheme for the Internet of things[Text].

16 M. Panwar, A. Kumar, “Security for IoT an effective DTLS with public certificates.”[Text]. // International conference on advances in Computer Engineering and application (ICACEA), - 2015

17 F. Santoso, N. Vun, “Securing IoT for Smart Home System .”[Text]. // International Symposium on Consumer Electronics (ISCE) Securing, - 2015 IEEE

18 J. Lee, W. Lin, Y. Huang, “A Lightweight Authentication Protocol for Internet of Things.”[Text]. // International Symposium on Next-Generation Electronics, ISNE 2014