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

До захисту допущено
Завідувач кафедри
_____ **Дмитро ЛАНДЕ**
(підпис)
«_____»_____2024 р

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою

«Системи, технології та математичні методи кібербезпеки»
зі спеціальності 125 «Кібербезпека»

на тему:

Виконав (-ла): здобувач вищої освіти 2 курсу, групи __ФБ-21мн__
(шифр групи)
Мороз Дмитро Валерійович
(прізвище, ім'я, по батькові) (підпис)

Керівник доцент кафедри інформаційної безпеки, канд. техн. наук, Гальчинський Леонід Юрійович
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Рецензент к.т.н., керівник Platform Security Lab, ТОВ "Самсунг РнД Інститут Україна", Сінельнікова Ольга Ігорівна
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по батькові) (підпис)

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

Київ – 2024 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Кафедра інформаційної безпеки

Рівень вищої освіти – другий (магістерський)

Спеціальність – 125 «Кібербезпека»

Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Дмитро ЛАНДЕ
(підпис)
«_____»_____2024 р.

ЗАВДАННЯ
на магістерську дисертацію здобувачу ступеня магістра

Морозу Дмитру Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи
____Моделювання та оцінка ризиків безпеки в контексті Edge-орієнтованих архітектур___,
керівник роботи
___Гальчинський Леонід Юрійович_____,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 10 _____квітня_ 2024р. № 1632-с_

2. Термін подання здобувачем вищої освіти роботи _08___червня__ 2024р.

3. Вихідні дані до роботи
_технічна документація, стандарти управління ризиками_____

4. Зміст роботи

1. Граничні обчислення як специфічний різновид розподілених обчислень

2. Модель та її компоненти
3. Експериментальні дослідження
4. Стартап проект

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація

6. Дата видачі завдання 23.02.2023

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Дослідження парадигми розподілених обчислень	01.02.2023 - 20.03.2023	виконано
2	Аналіз проблем кібербезпеки в Edge-орієнтованих середовищах	20.03.2023 - 01.06.2023	виконано
3	Робота над першим розділом, аналіз літературних джерел	01.06.2023 - 01.10.2023	виконано
4	Робота над другим розділом	01.10.2023 - 01.02.2024	виконано
5	Підготовка моделі, її реалізація та опис, обрахунок ризиків	01.02.2024 - 25.03.2024	виконано
6	Моделювання атак на модель	25.03.2024 - 15.05.2024	виконано
7	Розробка стартап проект та оформлення	15.05.2024 - 07.06.2024	виконано
8	Передзахист дипломної роботи	08.06.2024	виконано
9	Доопрацювання	08.06.2024 - 13.06.2024	виконано
10	Захист дипломної роботи	15.06.2024	виконано

Здобувач вищої освіти _____ Дмитро МОРОЗ
(підпис) (**Власне ім'я, ПРІЗВИЩЕ**)

Керівник роботи _____ Леонід ГАЛЬЧИНСЬКИЙ
(підпис) (**Власне ім'я, ПРІЗВИЩЕ**)

Реферат

Обсяг роботи 75 сторінок, 8 ілюстрацій, 3 таблиці, 22 джерел літератури.

Об'єктом дослідження є ризики безпеки в Edge-орієнтовних архітектурах.

Предметом дослідження є моделювання та оцінка ризиків безпеки в Edge-орієнтованих архітектурах.

Методи дослідження включають поєднання наявних фреймворків та методологій безпеки, зокрема NIST Risk Management Framework (RMF), застосованих до контексту edge обчислень.

Метою цієї роботи є розробка практичного рішення для підвищення безпеки в периферійних архітектурах за допомогою комплексного підходу до управління ризиками, інтегруючи агентні моделі для безпечної комунікації та безперервного моніторингу загроз.

Результати можуть бути використані для побудови більш безпечних периферійних обчислювальних систем, особливо в середовищах з пристроями Інтернету речей і потребами в обробці даних в реальному часі.

Ключові слова: граничні обчислення, ризики безпеки, захищені комунікації

Abstract

The work includes 75 pages, 8 illustrations, 3 tables, 22 bibliography references.

The object of the study is security risks in edge-oriented computing architectures.

The subject of the study is the modeling and evaluation of security risks in edge-oriented architectures.

Research methods include a combination of existing security frameworks and methodologies, specifically the NIST Risk Management Framework (RMF), applied to the context of edge computing.

The purpose of this work is to develop a practical solution for enhancing security in edge-oriented architectures through a comprehensive risk management approach, integrating agent-based models for secure communication and continuous threat monitoring.

The results can be utilized to build more secure edge computing systems, particularly in environments with IoT devices and real-time data processing needs.

Keywords: edge computing, security risks, secure communication

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ТЕРМІНІВ

MQTT - спрощений мережевий протокол, що працює на TCP/IP.

ChaCha20Poly1305 - це схема шифрування з автентифікацією, яка поєднує ChaChaTLS і Poly1305.

IoT - концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані давачі, а також програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами в автоматичному режимі, за допомогою використання стандартних протоколів зв'язку.

Industrial Internet of Things, IIoT — це система об'єднаних комп'ютерних мереж і підключених до них промислових (виробничих) об'єктів з вбудованими датчиками і програмним забезпеченням для збору та обміну даними, з можливістю віддаленого контролю і управління в автоматизованому режимі, без участі людини[22].

ВСТУП

Edge обчислення представляють собою зміну парадигми в інформаційних технологіях, обробляючи дані поблизу джерела, а не покладаючись виключно на централізовані хмарні сервіси. Цей підхід вирішує такі критичні проблеми, як затримка трафіку, обмеження пропускної здатності та необхідність обробки даних у реальному часі. Сфери застосування граничних обчислень доволі широкі і охоплюють промислову автоматизацію, розумні міста, автономні транспортні засоби, охорону здоров'я та роздрібну торгівлю.

Незважаючи на ці переваги, периферійні обчислення створюють значні проблеми з безпекою, так як їх розподілена природа робить вразливими до різних загроз, включаючи витік даних, атаки шкідливим програмним, фізичне втручання тощо. Традиційні моделі та фреймворки безпеки можуть покривати не всі чи не тією мірою як необхідно, унікальних ризиків у середовищах периферійних обчислень, що зумовлює необхідність розробки спеціальних стратегій оцінки та зменшення ризиків.

У даній роботі пропонується нова модель безпеки для Edge-орієнтованих архітектур, яка використовує агентний підхід для підвищення безпеки та ефективності. Модель включає безпечні протоколи зв'язку, надійні механізми автентифікації та безперервний моніторинг загроз. Застосовуючи NIST Risk Management Framework (RMF), можливо систематично оцінити ризики безпеки та застосовуючи запропоновану модель, зменшити їх. Експериментальні дані про використання ресурсів додатково підтверджують реалістичність і продуктивність запропонованої моделі в практичних сценаріях периферійних обчислень.

1 ГРАНИЧНІ ОБЧИСЛЕННЯ ЯК СПЕЦИФІЧНИЙ РІЗНОВИД РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ

Поява Edge (граничних, периферійних) обчислень, що характеризуються можливостями децентралізованої обробки та зберігання даних, здатна здійснити революцію в багатьох галузях і сферах застосування. Наближаючи обчислення до джерела даних, периферійні обчислення потенційно зменшують затримки, підвищують ефективність використання смуги пропускання та здійснюються посиленням конфіденційності даних. Однак ця зміна парадигми також створює новий набір проблем безпеки, які вимагають ретельного дослідження та інноваційних рішень.

Розподілена природа периферійних середовищ, які часто складаються з різнорідних пристроїв і мереж з обмеженими ресурсами, створює динамічний ландшафт загроз. Витоки даних, несанкціонований доступ і порушення конфіденційності становлять значні ризики, оскільки конфіденційна інформація збирається, обробляється і зберігається на кінцевих пристроях що мають доволі мало ресурсів для захисту. Поширення підключених пристроїв і залежність від потенційно вразливих протоколів зв'язку ще більше посилюють ці ризики. Крім того, фізична близькість периферійних пристроїв до потенційних зловмисників викликає занепокоєння щодо фізичного втручання, крадіжки та порушення навколишнього середовища.

1.1 Граничні обчислення

Граничні обчислення являють собою зміну в обчислювальній парадигмі, яка характеризується децентралізованою архітектурою, де обробка і зберігання даних відбувається ближче до джерела даних, часто на «краю»(Edge -- край) мережі. Це контрастує з традиційними моделями хмарних обчислень, де дані надсилаються для обробки в централізовані центри обробки даних[1,2]. Граничні обчислення зумовлені поширенням пристроїв IoT, потребою в аналітиці в режимі реального часу, а також обмеженнями пропускної здатності і затримок в мережі.

По суті, edge обчислення наближають обчислення до даних, а не дані до обчислень. Це дозволяє пришвидшити час відгуку, зменшити споживання пропускної здатності і підвищити рівень конфіденційності та безпеки, оскільки конфіденційні дані можна обробляти локально, не передаючи їх у хмару. Периферійні пристрої можуть варіюватися від простих датчиків і виконавчих механізмів до потужних серверів, здатних виконувати складні обчислення.

«Edge» в edge обчисленнях відноситься до різноманітних місць, включаючи локальні сервери, мережеві шлюзи, мікроцентри обробки даних, інколи і самі пристрої. Така розподілена природа забезпечує більшу гнучкість і масштабованість, оскільки ресурси можуть бути розгорнуті ближче до місця, де вони потрібні.

Однак децентралізація edge обчислень також створює унікальні виклики, такі як необхідність керувати гетерогенним(різного виду) набором пристроїв, забезпечувати послідовну безпеку в розподілених місцях і вирішувати проблеми, пов'язані з обмеженнями периферійних ресурсів. Ці виклики вимагають розробки нових підходів до оцінки та зменшення ризиків, пристосованих до специфічних характеристик периферійних середовищ.

Переваги та виклики граничних обчислень

Граничні обчислення пропонують альтернативу традиційним централізованим хмарним обчисленням, обіцяючи докорінно змінити способи обробки та

використання даних. Однак, як і будь-яка технологічна парадигма, існують як певні переваги, так і виклики, які необхідно ретельно розглянути[3].

Переваги:

- Зменшення затримок: Обробляючи дані ближче до їхнього джерела, периферійні обчислення значно зменшують затримку - час, необхідний для передачі даних між пристроями та хмарою. Це має вирішальне значення для додатків, які вимагають реагування в режимі реального часу, таких як промислова автоматизація, автономні транспортні засоби та телемедицина.
- Ефективність використання смуги пропускання: Граничні обчислення можуть зменшити навантаження на пропускну здатність мережі, фільтруючи і обробляючи дані локально, відправляючи в хмару тільки актуальну або оброблену інформацію. Це може призвести до значної економії коштів і підвищення продуктивності мережі
- Покращена конфіденційність і безпека: Завдяки edge обчисленням, конфіденційні дані можна обробляти і зберігати локально, що зменшує ризик їх витоку і несанкціонованого доступу під час передачі в хмару. Це особливо важливо для галузей із суворими правилами конфіденційності, таких як охорона здоров'я та фінанси.
- Підвищена надійність і відмовостійкість: Граничні обчислення можуть працювати незалежно від хмари, забезпечуючи безперервну роботу навіть під час відключень або збоїв в мережі. Це важливо для критично важливих додатків, які не терплять простоїв.
- Масштабованість і гнучкість: Граничні обчислення пропонують більшу масштабованість і гнучкість у порівнянні з централізованими хмарними архітектурами. Ресурси можна розгортати і керувати ними ближче до місця, де вони потрібні, забезпечуючи швидший час відгуку і більш ефективне використання ресурсів .

Виклики[4,5,6, 10]

- Вразливості безпеки: Розподілена природа граничних обчислень створює ширшу поверхню для атак. Кінцеві пристрої можуть мати обмежені можливості захисту в порівнянні з централізованими хмарними серверами, що вимагає додаткових заходів для захисту від несанкціонованого доступу, витоку даних і зараження шкідливим програмним забезпеченням.
- Обмеженість ресурсів: Граничні пристрої часто мають обмежену обчислювальну потужність, обсяг пам'яті та енергетичні ресурси порівняно з хмарними серверами. Це вимагає ретельної оптимізації алгоритмів і додатків для забезпечення ефективного використання ресурсів.
- Гетерогенність та інтероперабельність: Граничні середовища часто складаються з різноманітних пристроїв, операційних систем і протоколів зв'язку. Забезпечення безперебійної взаємодії та сумісності між цими різнорідними компонентами може бути складним завданням
- Управління та обслуговування: Управління та обслуговування великої кількості розподілених граничних пристроїв може бути складним завданням, що вимагає складних інструментів і автоматизації для моніторингу продуктивності, оновлення програмного забезпечення та усунення несправностей.
- Вартість і складність: Хоча периферійні обчислення можуть зменшити витрати, пов'язані з хмарними технологіями, початкові інвестиції в обладнання, програмне забезпечення та інфраструктуру можуть бути значними. Крім того, складність управління розподіленим периферійним середовищем може створити проблеми для організацій з обмеженими ресурсами.

Сфера застосування

Універсальність edge обчислень призвела до їх впровадження в широкий спектр галузей і додатків. Наближаючи обчислення до джерела даних, граничні обчислення уможливлюють нові рішення, які раніше були непрактичними або

неможливими через затримки, обмеження пропускної здатності або конфіденційності [7].

- Промисловий IoT (IIoT): Граничні обчислення відіграють ключову роль в IIoT, де обробка даних в режимі реального часу і прийняття рішень мають вирішальне значення для оптимізації промислових процесів, прогнозування технічного обслуговування і виявлення аномалій. Периферійні пристрої можуть аналізувати дані датчиків машин і обладнання, виявляти потенційні проблеми і запускати коригувальні дії, не покладаючись на хмарне підключення.
- Розумні міста: Граничні обчислення розширюють можливості ініціатив «розумного міста», дозволяючи в режимі реального часу аналізувати дані з різних датчиків і пристроїв, розгорнутих по всьому міському середовищу. Ці дані можна використовувати для управління дорожнім рухом, оптимізації енергоспоживання, громадської безпеки та моніторингу навколишнього середовища, підвищуючи ефективність і придатність міст для життя.
- Автономні транспортні засоби: Граничні обчислення необхідні для безпечної та ефективної роботи автономних транспортних засобів. Бортові периферійні пристрої можуть обробляти дані з камер та радарів в режимі реального часу, дозволяючи транспортному засобу приймати рішення щодо навігації, об'їзду перешкод і управління дорожнім рухом якнайшвидше.
- Охорона здоров'я: Граничні обчислення можуть уможливити віддалений моніторинг пацієнтів, діагностику в реальному часі та персоналізовану медицину. Периферійні пристрої можуть збирати та аналізувати дані про пацієнтів з натільних датчиків і медичних пристроїв, надаючи медичним працівникам цінну інформацію для раннього втручання та оптимізації лікування.
- Доповнена і віртуальна реальність (AR/VR): Граничні обчислення можуть значно підвищити продуктивність і швидкість реагування додатків AR/VR, перекладаючи обчислювально інтенсивні завдання на периферійні сервери. Це зменшує затримку і забезпечує більш захоплюючий і реалістичний досвід.

- Фінансові послуги: Граничні обчислення можуть сприяти швидшому та безпечнішому проведенню фінансових транзакцій завдяки виявленню шахрайства в режимі реального часу, оцінці ризиків та моніторингу дотримання нормативних вимог, в той же час обробляючи дані про транзакції локально, знижуючи ризик витоку даних.
- Сільське господарство: Граничні обчислення можуть трансформувати сільське господарство, дозволяючи застосовувати методи точного землеробства, моніторингу посівів і управління тваринництвом.

1.2 Ризики безпеки в Edge середовищах

Розподілена і гетерогенна природа edge обчислень створює унікальний ландшафт загроз, який суттєво відрізняється від традиційних моделей централізованих обчислень. Edge середовища часто складаються з численних пристроїв з різним рівнем безпеки, що працюють в різних фізичних місцях і мережевих умовах. Це створює більш широку поверхню для атак і піддає системи більш широкому спектру потенційних загроз[15-23].

Унікальний ландшафт загроз

- Розподілена поверхня атаки: На відміну від централізованих хмарних середовищ, периферійні обчислення включають безліч пристроїв і кінцевих точок, кожна з яких є потенційною точкою входу для зловмисників. Така розподілена природа ускладнює захист кожного пристрою і з'єднання, збільшуючи ризик несанкціонованого доступу і витоку даних.
- Обмеженість ресурсів: Багато периферійних пристроїв обмежені в ресурсах, з обмеженою обчислювальною потужністю, пам'яттю і ємністю сховища. Це може ускладнити впровадження надійних заходів безпеки, таких як шифрування або системи виявлення вторгнень, без шкоди для продуктивності.
- Ризики пов’язані з фізичною безпекою: Граничні пристрої часто розгортаються в неконтрольованих середовищах, що робить їх вразливими до фізичного втручання, крадіжки або пошкодження.
- Гетерогенність: периферійні середовища часто складаються з різноманітних пристроїв, операційних систем і програмних додатків, кожен з яких має власні вразливості безпеки. Така гетерогенність ускладнює застосування узгоджених політик і конфігурацій безпеки у всьому середовищі.
- Мережеве підключення: Граничні пристрої покладаються на мережеве підключення для зв'язку один з одним і з хмарою. Ця залежність робить їх вразливими до мережевих атак, таких як атаки типу «зловмисник посередині», атаки типу «відмова в обслуговуванні» і перехоплення даних.

Безпека та конфіденційність даних

Безпека та конфіденційність даних є першочерговими проблемами в контексті edge обчислень, оскільки конфіденційні дані часто збираються, обробляються та зберігаються на кінцевих пристроях. Однак розподілена природа периферійних середовищ і обмеженість ресурсів периферійних пристроїв можуть ускладнити реалізацію адекватних заходів безпеки.

Витоки даних: Периферійні пристрої вразливі до витоку даних через їх вразливість до фізичних та мережевих атак. Якщо пристрій скомпрометовано, зловмисники можуть отримати доступ до конфіденційних даних, що зберігаються на ньому або передаються мережею.

Несанкціонований доступ: Граничні пристрої можуть не мати надійних механізмів автентифікації та авторизації, що робить їх вразливими до несанкціонованого доступу. Зловмисники можуть використовувати вразливості в пристрої або його програмному забезпеченні, щоб отримати доступ і контроль, потенційно компрометуючи дані та сервіси.

Конфіденційність даних: Граничні обчислення викликають занепокоєння щодо конфіденційності даних, оскільки конфіденційні дані можуть збиратися і оброблятися без відома або згоди користувача. Крім того, дані можуть передаватися стороннім постачальникам послуг або рекламодавцям, що викликає занепокоєння щодо права власності на дані та контролю над ними.

Цілісність даних: Розподілена природа периферійних обчислень може ускладнити забезпечення цілісності даних, оскільки вони можуть бути репліковані та змінені на різних пристроях. Це може призвести до невідповідностей і помилок, що ставить під загрозу точність і надійність даних.

Витік даних: Периферійні пристрої можуть ненавмисно допустити витік конфіденційних даних через незахищені канали зв'язку або незашифровані сховища. Це може призвести до того, що особиста інформація, конфіденційні бізнес-дані або інтелектуальна власність стануть доступними стороннім особам.

Щоб вирішити ці проблеми безпеки та конфіденційності даних, середовища периферійних обчислень потребують надійних заходів безпеки, таких як шифрування, контроль доступу, анонімізація даних та безпечні протоколи зв'язку. Крім того, організаціям необхідно впровадити чіткі політики і процедури управління даними, щоб забезпечити дотримання правил конфіденційності та захистити конфіденційну інформацію.

Висновки за розділом 1

Граничні обчислення розвивають інформаційні технології, обробляючи дані ближче до джерела, пропонуючи такі переваги, як зменшення затримок, посилення конфіденційності та підвищення надійності. Цей зсув зумовлений зростанням кількості пристроїв Інтернету речей, потребами в аналітиці в режимі реального часу та мережевими обмеженнями. Граничні обчислення застосовуються у сферах промислової автоматизації, розумних міст, автономних транспортних засобів, охорони здоров'я та роздрібної торгівлі, забезпечуючи моніторинг у реальному часі, управління трафіком, дистанційну діагностику та персоналізований клієнтський досвід.

Однак периферійні обчислення створюють унікальні виклики для безпеки, включаючи ризики безпеки даних і конфіденційності, вразливості пристроїв і мереж, а також загрози фізичній безпеці. Розподілена природа периферійних пристроїв та обмежені можливості захисту роблять їх мішенями для кібератак, шкідливого програмного забезпечення та фізичного втручання.

2 МОДЕЛЬ ТА ЇЇ КОМПОНЕНТИ

2.1. Загальні риси

Основна мета запропонованої моделі для Edge-орієнтованих середовищ - забезпечення безпечної, легкої та адаптивної основи для розгортання агентів на периферійних пристроях. Ці агенти призначені для виконання різноманітних завдань, таких як збір даних, аналіз і прийняття рішень, при цьому мінімізуючи використання ресурсів. Використовуючи модель безпеки з нульовою довірою, модель гарантує, що зв'язок між агентами і центральним edge вузлом постійно аутентифікується і шифрується, забезпечуючи захист конфіденційних даних і захист від несанкціонованого доступу.

Модель враховує розподілену природу периферійних мереж, дозволяючи агентам працювати автономно, підтримуючи при цьому безпечний зв'язок з edge вузлом. Крім того, акцент на ефективність використання ресурсів дозволяє агентам ефективно функціонувати на обмежених пристроях, максимізуючи потенціал периферійних обчислень у різних сферах, включаючи Інтернет речей (IoT), промислову автоматизацію та розумні міста.

Основним принципом цієї моделі є прийняття архітектури нульової довіри. Такий підхід суттєво відрізняється від традиційних моделей безпеки, які зазвичай покладаються на захист «по периметру». У цих моделях все, що знаходиться всередині мережі, як правило, користується довірою, в той час як все, що знаходиться за її межами, вважається потенційною загрозою. Нульова довіра, з іншого боку, виходить з припущення, що за замовчуванням нікому не можна довіряти - ні користувачам, ні пристроям, ні навіть об'єктам всередині мережі. Кожен запит на доступ, незалежно від його походження, повинен бути перевірений, перш ніж він буде дозволений.

Така зміна перспективи має вирішальне значення для середовищ edge обчислень, де пристрої часто розосереджені і підключені до мереж з різним рівнем безпеки.

Застосовуючи цей принцип, модель робиться більш надійною та захищеною від широкого спектру загроз. Навіть якщо пристрій у мережі скомпрометований, потенційна шкода обмежена, тому що цей пристрій може отримати доступ лише до певних ресурсів, необхідних йому для роботи. Якщо він спробує зробити щось інше, його буде виявлено і зупинено.

2.2. Агент

Агент - це легкий, високоефективний програмний компонент, призначений для роботи на периферійних пристроях з обмеженими обчислювальними ресурсами та пам'яттю. Його основна роль полягає у забезпеченні безпечного зв'язку з edge вузлом (що діє як MQTT брокер), обробці даних та виконанні локальних завдань з виявлення загроз (опціонально).

Агент є критично важливим компонентом запропонованого edge-орієнтованого середовища. Він працює на обмежених пристроях, таких як датчики Інтернету речей(IoT) або невеликі периферійні обчислювальні пристрої, і виконує кілька ключових функцій для забезпечення безпеки та ефективності системи. Агент має бути розроблений таким чином, щоб мінімізувати навантаження на процесор і пам'ять, ефективно працювати в умовах обмежених ресурсів периферійних пристроїв.

Функціональність агента

Функціональність агента можна розділити на кілька основних напрямків:

Комунікація

У середовищі периферійних обчислень ефективний та безпечний зв'язок між агентами та центральними вузлами має пріоритетне значення. Для забезпечення цього використовується протокол MQTT: легкий протокол обміну повідомленнями, розроблений для пристроїв з обмеженими можливостями.

Роль клієнта

Агент працює як клієнт MQTT, підключаючись до брокера MQTT, розміщеного на edge вузлі. Кожен агент має унікальну комбінацію логіна та пароля для початкової автентифікації. Це гарантує, що тільки авторизовані агенти можуть встановити з'єднання.

Механізми автентифікації

При першому підключенні агент використовує своє унікальне ім'я користувача та пароль для автентифікації з брокером MQTT. Для підвищення безпеки система використовує автентифікацію на основі токенів. Після успішної початкової автентифікації агент періодично отримує короткочасний токен від визначеного топіку MQTT. Цей токен має регулярно оновлюватись, для підтримання дійсності сесії, зменшуючи ризик довготривалого витоку облікових даних.

Підписки на топіки

Агент підписується на кілька спеціальних *топіків*, для отримання команд, оновлень та інформацію про загрози. Нижче наведено приклад тем які можуть включатись, але не обмежуватись:

- commands/{clientId}: для отримання оперативних команд.
- updates/{clientId}: для отримання оновлень програмного забезпечення або змін конфігурації.
- threat_intel/{clientId}: для отримання останньої інформації про загрози.

Приклад потоку повідомлень

- Початкове підключення: агент підключається до брокера MQTT, використовуючи свої унікальні облікові дані.
- Отримання токену: брокер публікує токен у топіку *token_for/{clientId}*, на яку підписаний агент.
- Оперативні команди: агент підписується на *commands/{clientId}*, щоб отримувати інструкції від граничного вузла.
- Регулярне оновлення токенів: періодично агент надсилає запит на *token_refresh/{clientId}*. Брокер відповідає новим токеном, який публікується *на token_for/{clientId}*.

Використовуючи MQTT, агент забезпечує ефективний та безпечний зв'язок, що є життєво важливим для підтримки цілісності та швидкості реагування середовищ периферійних обчислень.

Шифрування

Безпека є критично важливим аспектом периферійних обчислень, де конфіденційні дані повинні бути захищені від несанкціонованого доступу та підробки. Алгоритм шифрування ChaCha20Poly1305 був обраний за його продуктивність і переваги безпеки, забезпечуючи як шифрування, так і автентифікацію.

- *Процес шифрування*

ChaCha20Poly1305 — сучасний алгоритм шифрування, відомий своєю ефективністю та безпекою. Він поєднує в собі потоковий шифр ChaCha20 для шифрування та MAC Poly1305 для автентифікації, забезпечуючи цілісність та конфіденційність даних.

- *Стратегії управління ключами*

Початкове надання ключів: ключі безпечно доставляються агенту через апаратний модуль безпеки (HSM) або аналогічний захищений позасмуговий механізм. Таке початкове надання ключів гарантує, що ключі шифрування отримають лише авторизовані агенти.

Ротація ключів(за можливості): для посилення безпеки ключі шифрування періодично оновлюються. Це зменшує ризик компрометації ключів і гарантує, що навіть якщо ключ буде вразливим, час його вразливості буде обмеженим.

- *Приклад реалізації:*
    1. Шифрування: коли агент надсилає повідомлення, він шифрує корисне навантаження за допомогою алгоритму ChaCha20 і генерує тег автентифікації за допомогою Poly1305.
    2. Передача: зашифроване корисне навантаження та тег автентифікації надсилаються брокеру MQTT.
    3. Розшифровка: після отримання повідомлення, граничний вузол використовує спільний ключ для розшифровки корисного навантаження і перевірки тегу автентифікації, забезпечуючи цілісність даних.

Використання ChaCha20Poly1305 гарантує, що конфіденційна інформація, яка передається між агентом та периферійним вузлом, захищена від підслуховування та фальсифікації. Це має вирішальне значення для підтримання надійності периферійного обчислювального середовища.

Виявлення загроз

В контексті edge обчислень, механізми виявлення загроз повинні бути розроблені таким чином, щоб ефективно працювати в умовах обмежених ресурсів.

- *Виявлення аномалій*

Виявлення аномалій є критично важливим аспектом виявлення загроз. Воно передбачає моніторинг системи для виявлення незвичайних шаблонів або поведінки, які можуть вказувати на загрозу безпеці. Враховуючи обмеженість ресурсів периферійних пристроїв, агент використовує полегшений підхід:

1. **Сканування портів**:
    - Мета: виявлення несанкціонованих мережевих підключень або служб, запущених на пристрої.
    - Реалізація: агент періодично сканує заздалегідь визначений діапазон локальних портів. Будь-які не санкціоновано відкриті порти позначаються як потенційні загрози безпеці.
    - Міркування щодо ресурсів: сканування портів оптимізовано для мінімізації використання процесора і пам'яті, воно виконується з низьким пріоритетом, щоб не порушувати нормальну роботу.
2. **Аналіз мережевого трафіку (опціонально)**:

- Мета: виявити аномальні шаблони мережевого трафіку, які можуть свідчити про порушення безпеки.
- Реалізація: агент відстежує вихідний і вхідний мережевий трафік, шукаючи незвичайні сплески або з'єднання з відомими зловмисними IP-адресами.
- Обмеження щодо ресурсів: аналіз трафіку виконується через невеликі проміжки часу, щоб зменшити споживання ресурсів. Аналізуються лише

зведені дані (наприклад, кількість з'єднань, обсяг даних), щоб зробити процес легким.

3. **Моніторинг цілісності файлів (опціонально)**

Якщо кінцевий пристрій має достатньо ресурсів, агент може виконувати моніторинг цілісності файлів для виявлення несанкціонованих модифікацій критично важливих системних файлів:

- ***Обчислення базового хешу***:

Мета: встановити незмінність критично важливих системних файлів.

Реалізація: агент обчислює та зберігає криптографічні хеші (наприклад, SHA-256) критично важливих файлів під час початкової фази налаштування.

- ***Періодичні перевірки цілісності***:

Мета: Виявлення несанкціонованих змін у критично важливих файлах.

Реалізація: Через певні проміжки часу агент перераховує хеші критично важливих файлів і порівнює їх з базовим значенням. Будь-які розбіжності позначаються прапорцями і повідомляються граничному вузлу.

- **Ресурсні міркування**:

Частоту перевірок цілісності можна регулювати залежно від наявних ресурсів. Перевірки можуть бути розподілені в часі, щоб мінімізувати вплив на ресурси.

Втім через обмеженість ресурсів деяких кінцевих пристроїв впровадження комплексного виявлення загроз може бути складним завданням. Ось кілька стратегій, які допоможуть вирішити ці проблеми:

Визначення пріоритетів: зосередження на найбільш важливих загрозах, які становлять найбільший ризик.

Кастомізація: налаштування функції виявлення загроз на основі конкретних можливостей і сценаріїв використання пристрою.

Розвантаження: там, де це можливо, перекладення складних завдань виявлення на більш потужні пристрої або edge вузли.

4. **Додаткові міркування**:

Забезпечення надійності та безпеки агента виходить за рамки базової функціональності. Декілька додаткових міркувань є важливими для підтримки цілісності агента та забезпечення безперебійного оновлення.

Secure boot є критично важливою функцією безпеки, яка забезпечує цілісність програмного забезпечення агента під час запуску, запобігаючи завантаженню несанкціонованого або шкідливого програмного забезпечення.

Призначення є запобігання запуску несанкціонованого або підробленого програмного забезпечення. Реалізація можлива через перевірку завантажувача(bootloader'a): Агент використовує захищений завантажувач, який перевіряє цілісність образу програмного забезпечення перед його завантаженням. Ця перевірка зазвичай включає перевірку криптографічного підпису. Або створивши ланцюжок довіри від апаратного забезпечення до операційної системи, гарантуючи, що кожен компонент буде перевірений перед виконанням.

Оновлення OTA(Over-the-Air) дозволяють агенту безпечно та ефективно отримувати оновлення програмного забезпечення та дані про загрози. Мета — гарантувати, що агент завжди працює з останньою версією програмного забезпечення та має актуальні дані про загрози. Пропонований варіантів реалізації:

Доставка оновлень: оновлення доставляються безпечно через протокол MQTT. Пакет оновлень шифрується за допомогою ChaCha20Poly1305 для забезпечення конфіденційності.

Перевірка цілісності: перед застосуванням оновлення агент перевіряє його цілісність за допомогою криптографічних контрольних сум. Тільки перевірені оновлення застосовуються для запобігання втручанню.

Пропоновані мови для реалізації

Вибір мови програмування для агента є ключовим рішенням, яке суттєво впливає на продуктивність, використання ресурсів та його загальні можливості.

Двома перспективними кандидатами на цю роль є Rust і MicroPython, кожна з яких пропонує унікальне поєднання переваг і міркувань.

Rust: сучасна мова системного програмування, відома своєю зосередженістю на продуктивності, безпеці та ретельному управлінні пам’яттю[8].

Продуктивність: абстракції з нульовою вартістю та відсутність збирача сміття(garbage collector) забезпечують надзвичайну ефективність Rust. Вона часто конкурує або перевершує традиційні системні мови, такі як C та C++, за швидкістю та споживанням пам'яті, що робить її привабливим вибором для кінцевих пристроїв з обмеженими ресурсами.

Безпека: модель власності Rust у поєднанні з суворими перевірками компілятора допомагає усунути поширені помилки програмування, такі як нульові посилання на вказівники, race condition та переповнення буферу. Такий надійний підхід до безпеки значно підвищує надійність програмного забезпечення агента.

Ефективність використання пам'яті: явне керування пам'яттю в Rust надає можливість детального контролю над розподілом ресурсів. Це має вирішальне значення в edge середовищах, де пам'ять часто є “розкішшю”.

Крос-платформна сумісність: Rust легко компілюється на широкий спектр платформ, включаючи як малопотужні мікроконтролери, так і високопродуктивні сервери. Ця універсальність є перевагою для розробки агентів, призначених для різних периферійних пристроїв.

Вбудовані(embedded) можливості: Rust має багатий набір функцій та бібліотек, пристосованих для вбудованої розробки. Він навіть дозволяє компілювати агента безпосередньо як прошивку для мікроконтролерів, якщо такий рівень інтеграції є необхідним.

MicroPython: Спрощена та оптимізована версія мови програмування Python, ретельно розроблена для мікроконтролерів та середовищ з обмеженими ресурсами.

- Простота використання: MicroPython ставить на перше місце досвід розробника, маючи чіткий і лаконічний синтаксис, що сприяє швидкій розробці. Розгалужена екосистема бібліотек спрощує виконання поширених

завдань, таких як мережева комунікація, шифрування та взаємодія з сенсорами.

- Швидка розробка: інтерпретована природа MicroPython забезпечує швидкі ітерації та експерименти, дозволяючи розробникам швидко тестувати та вдосконалювати свій код.
- Багата екосистема: MicroPython має велику колекції бібліотек, надаючи готові рішення для безлічі завдань.
- Інтерпретована природа виконання: при тому що інтерпретована природа спрощує розробку, вона може призвести до повільнішого виконання порівняно з компільованими мовами, такими як Rust, що може бути важливим фактором для критично важливих до продуктивності додатків.

Огляд діаграм станів і процесів

Діаграми станів (рис. 2.2) та процесів(рис. 2.1) для агента ілюструють ключові стани, переходи та процеси, що визначають його роботу у edge-орієнтованому середовищі. Ці діаграми надають чітке і стисле уявлення про поведінку агента, включаючи те, як він обробляє MQTT-зв'язок, управління токенами, шифрування і потенційне виявлення загроз.

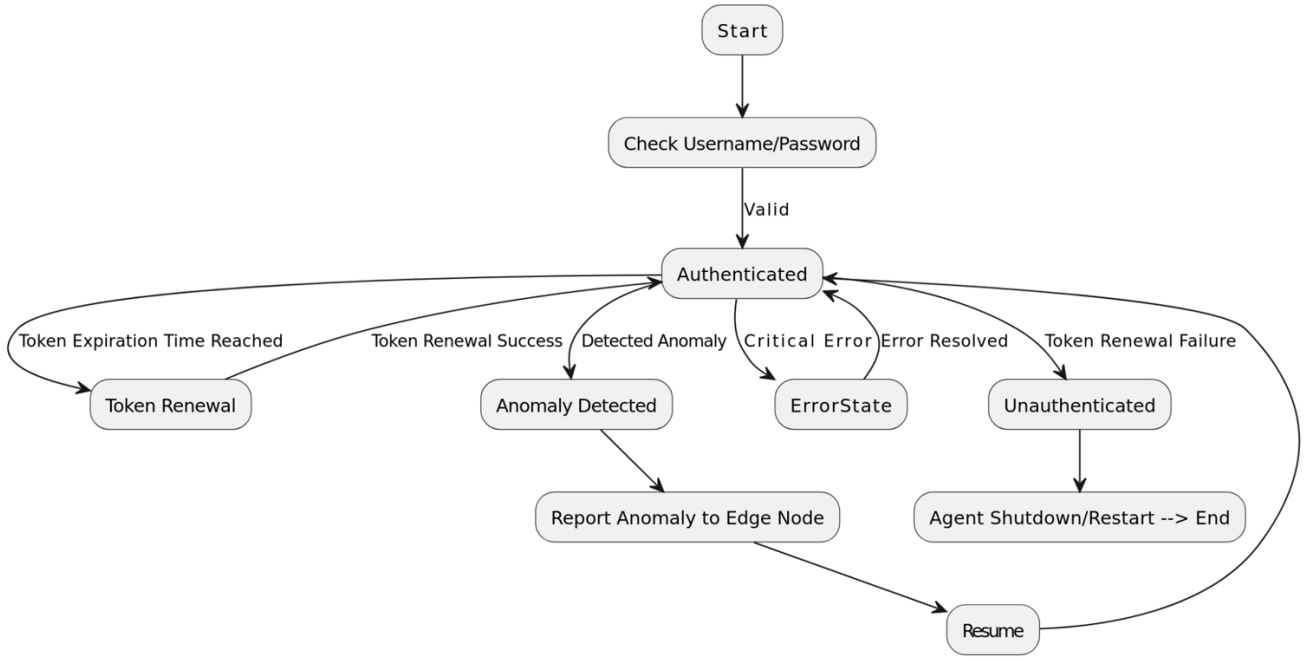

Рисунок 2.1 – Діаграми процесів агента

Нижче наведено детальний опис діаграм:

- *Старт*: початковий стан агента при запуску.
- *Перевірка ім'я користувача/пароля*: агент намагається автентифікуватися з брокером MQTT, використовуючи ім'я користувача та пароль.

  - Якщо вони дійсні, відбувається перехід до стану «Автентифіковано».

  - Якщо невірно, повторюється спроба або відбувається обробка помилки.
- *Автентифіковано*: агента успішно автентифіковано і він працює у звичайному режимі.

  - *Досягнуто час закінчення терміну дії токена*: при завершенні терміну дії токена, агент переходить у стан «Поновлення токена».

  - *Виявлена аномалія*: при виявлені аномалії, агент переходить у стан «Виявлена аномалія».

  - *Критична помилка*: перехід у стан «Помилка» при виявленні критичної помилки.

  - *Token Renewal Success* (Успішне поновлення токену): повертається до стану «Автентифіковано» після успішного оновлення токена.

  - *Помилка вирішена*: повертається до стану «Автентифіковано» після усунення помилки.
- *Поновлення токену*: агент намагається оновити свій токен.

  - У разі успіху повертається до стану «Автентифіковано».

  - У разі невдачі переходить до стану «Неавтентифікований».
- Виявлено аномалію: якщо виявлено аномалію, агент повідомляє про неї edge вузол.

  - Повідомлення про аномалію edge вузлу: це дія, яка виконується у цьому стані перед потенційним переходом до «Відновити».
- Стан помилки (Error State): агент переходить у цей стан при виникненні критичної помилки.

  - Вирішення помилки: якщо помилку вирішено, він повертається до стану «Автентифіковано».

- Неавтентифіковано: якщо помилку не вдається усунути, це може призвести до деавторизації та подальшого вимкнення або перезапуску.

• Неавтентифіковано: агент не пройшов автентифікацію, як правило, після невдалого оновлення токенів або помилки, яку неможливо виправити.

- Вимкнення/перезапуск агента: фінальний стан, коли агент вимикається або перезапускається.

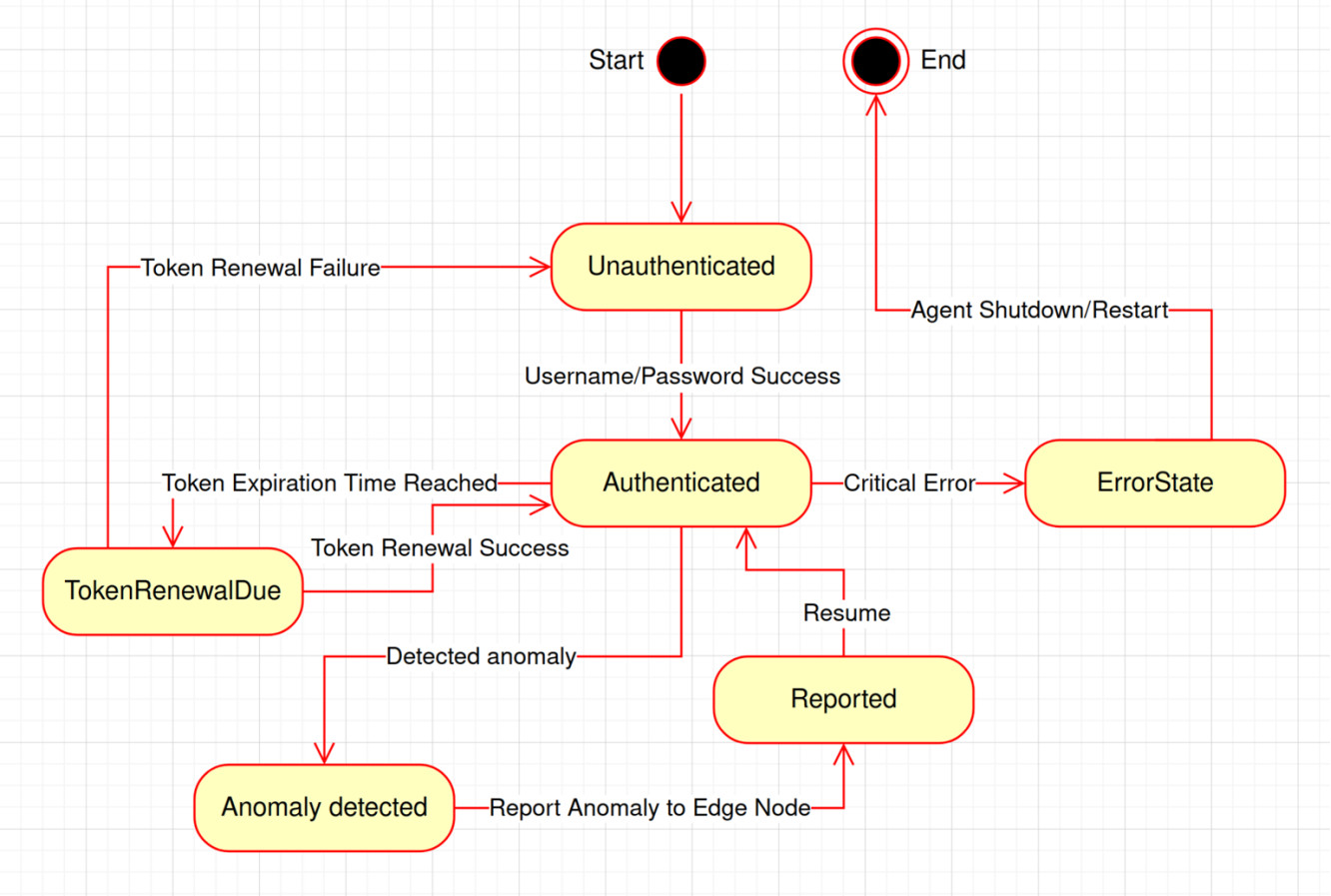

Рисунок 2.2 – Діаграми станів агента

2.3 Edge вузол (MQTT-брокер та управління безпекою)

Edge вузол є ключовим компонентом у запропонованій моделі, який виконує кілька ролей для забезпечення безпечного, ефективного та надійного зв'язку між агентом та мережею. Він діє як брокер MQTT, керує протоколами безпеки, відстежує дії та забезпечує дотримання політик.

MQTT-брокер

Брокер MQTT є основним компонентом edge вузла, який відповідає за управління MQTT-з'єднаннями, маршрутизацію повідомлень між клієнтами та забезпечення автентифікації. Його основні функції включають

1. Управління з'єднаннями: брокер обробляє з'єднання від декількох агентів, забезпечуючи унікальну ідентифікацію та автентифікацію кожного клієнта. Сюди входить ведення списку активних клієнтів та їхніх станів.
2. Маршрутизація токенів: MQTT працює за моделлю «публікація-підписка»(pub-sub), де повідомлення надсилаються до тем. Брокер направляє ці повідомлення клієнтам, підписаним на певні теми, забезпечуючи ефективну і точну доставку.
3. Аутентифікація: брокер перевіряє облікові дані клієнта під час процесу з'єднання. Початкова автентифікація використовує ім'я користувача та пароль, тоді як поточні сеанси підтримуються за допомогою автентифікації на основі токенів. Цей дворівневий підхід підвищує безпеку, забезпечуючи безперервну перевірку особи клієнта.
4. Масштабованість та продуктивність: для ефективної роботи в граничних середовищах, брокер має бути розроблений для обробки великих обсягів повідомлень з мінімальними затримками і споживанням ресурсів. Для підтримки продуктивності треба використовувати такі методи, як асинхронна обробка та ефективні структури даних.

Служба генерації токенів

Служба генерації токенів відповідає за випуск, оновлення та перевірку короткочасних токенів, які використовуються для підтримки аутентифікованих сесій між агентом та брокером MQTT. Цей сервіс гарантує, що тільки авторизовані пристрої можуть взаємодіяти через периферійний вузол. Процес включає в себе:

- Видачу токенів: після початкового підключення та успішної автентифікації, граничний вузол генерує короткочасний токен для агента. Цей токен публікується в певній темі (наприклад, “token_for/{clientId}”), де агент може його отримати.
- Оновлення токена: для підтримки безпечного сеансу, граничний вузол періодично видає нові токени до того, як закінчиться термін дії поточного. Це передбачає підписку на тему для оновлення токенів (наприклад, “token_refresh/#”) щоб визначити, яким клієнтам потрібен новий токен, а потім опублікувати новий токен у відповідній темі.
- Валідація: коли агент надсилає дані, він включає поточний токен. Граничний вузол перевіряє цей токен, порівнюючи його з відомим дійсним токеном. Якщо токен дійсний, дані обробляються; в іншому випадку агенту надсилається повідомлення про недійсний токен.
- Управління чергою: граничний вузол використовує чергу для управління запитами на генерацію токенів, забезпечуючи ефективну обробку і своєчасну видачу токенів.

Впровадження політики безпеки

Політики безпеки є критично важливими для підтримки цілісності та конфіденційності зв'язку між агентом та edge вузлом. Граничний вузол забезпечує дотримання наступних політик:

- Контроль доступу: брокер обмежує доступ до тем MQTT на основі облікових даних та ролей клієнта. Це гарантує, що агенти мають доступ лише до тем, необхідних для їхньої роботи, дотримуючись принципу найменших привілеїв.
- Безпека на рівні тем: темам можуть бути призначені різні рівні безпеки, які визначають, хто може публікувати або підписуватися на них.
- Оновлення політики: політики безпеки періодично переглядаються і оновлюються, щоб адаптуватися до нових загроз і вразливостей. Ці оновлення надсилаються на edge вузли, гарантуючи, що найновіші заходи безпеки завжди будуть на місці.

Моніторинг підозрілої активності (опціонально)

Граничний вузол безперервно відстежує використання токенів та дані що йдуть від клієнта, для виявлення і реагування на підозрілі дії. Це включає в себе:

1. Виявлення аномалій: граничний вузол відстежує використання токенів, частоту повідомлень і шаблони для виявлення аномалій. Наприклад, кілька невдалих перевірок токенів або незвичайна частота повідомлень можуть вказувати на порушення безпеки або несправність.
2. Реагування на інциденти: при виявленні підозрілої активності периферійний вузол може виконати заздалегідь визначені дії, такі як реєстрація події, оповіщення адміністраторів або тимчасове блокування підозрілого клієнта.
3. Логування та аналіз: всі дії та виявлені аномалії записуються в журнал для подальшого аналізу. Ці журнали допомагають зрозуміти вектори атак, покращити заходи безпеки та забезпечити дотримання політик безпеки.

Логування

Ефективне ведення журналу подій(логів, з англ. logs) має важливе значення для моніторингу, налагодження та аудиту. Edge вузол має реалізовувати надійний механізм ведення журналу для фіксації важливих подій та дій:

- Журналювання подій: граничний вузол реєструє критичні події, такі як спроби з'єднання, успіхи і невдачі аутентифікації, генерація і перевірка токенів, а також деталі маршрутизації повідомлень.
- Журнали безпеки: спеціальні журнали ведуться для подій, пов'язаних з безпекою, включаючи підозрілі дії, порушення політики та інциденти безпеки.
- Зберігання та захист: Журнали надійно зберігаються з дотриманням відповідних політик зберігання. Це гарантує, що відповідні журнали доступні, коли це необхідно, а старі або неактуальні журнали архівуються або видаляються.

Командування та контроль

Граничний вузол також має можливість надсилати команди або оновлення агенту . Ця функція використовується для:

- Оновлення програмного забезпечення: Оновлення по повітрю (OTA) можуть бути надіслані агенту Rust для безпечного оновлення його програмного забезпечення або бази даних розвідки загроз.
- Зміни конфігурації: edge вузол може надсилати зміни конфігурації агенту Rust, такі як оновлення підписаних тем або зміна операційних параметрів.
- Обробка інцидентів: У разі виявлення аномалій або порушень політики, граничний вузол може надсилати команди агенту для виконання

коригувальних дій, таких як ініціювання безпечного перезавантаження або зміна своєї поведінки для пом'якшення загроз.

Таким чином, роль периферійного вузла в цій моделі є всеосяжною і охоплює управління з'єднаннями, забезпечення безпеки, моніторинг активності та виконання команд. Забезпечуючи безпечний і ефективний зв'язок, периферійний вузол допомагає підтримувати цілісність і функціональність всієї системи, що відповідає принципам архітектури нульової довіри і периферійних обчислень.

Діаграма станів та процесів для Edge вузла

Наведені нижче діаграми ілюструють потік і переходи станів Edge вузла, забезпечуючи чітку візуалізацію його поведінки в управлінні комунікаціями MQTT, управлінні токенами, дотриманні безпеки і виявленні аномалій.

Діаграма станів для граничного вузла

Діаграма станів (рис. 2.3) для граничного вузла відображає різні стани, в яких він може перебувати протягом свого життєвого циклу, а також переходи між цими станами. Це допомагає зрозуміти, як граничний вузол реагує на різні події та умови.

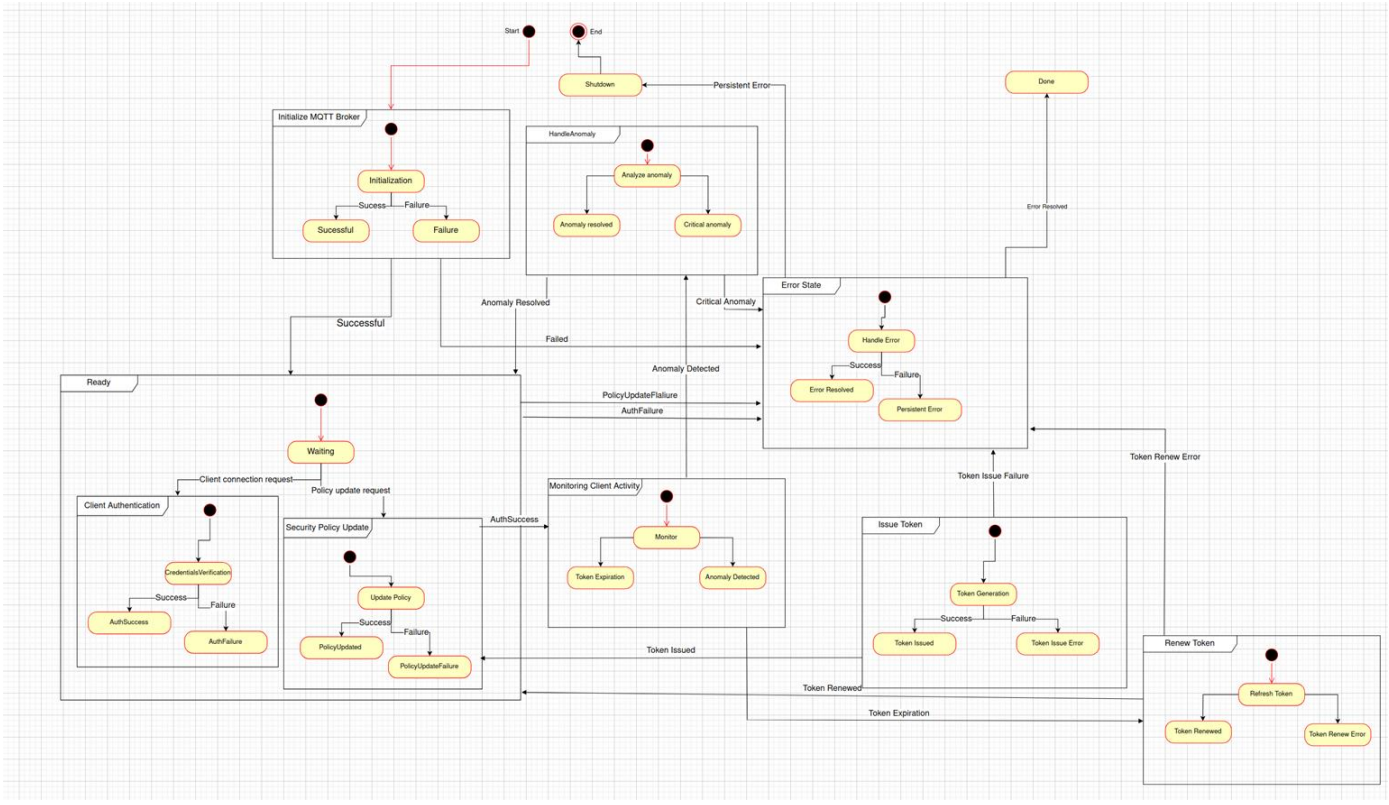

Рисунок 2.3 – Діаграми станів агента

Опис діаграми:

1. Старт: початковий стан, з якого починає свою роботу реберний вузол.

2. Ініціалізація MQTT-брокеру: граничний вузол ініціалізує MQTT-брокер, встановлюючи необхідні конфігурації.

- Ініціалізація: виконується ініціалізація брокера.

- Успіх ініціалізації: при успішній ініціалізації, стан переходить до «Готовий».

- Неуспішна ініціалізація: при невдалій ініціалізації, стан переходить у «Стан помилки».

3. Готовність: граничний вузол готовий приймати клієнтські з'єднання та керувати MQTT-зв'язком.

- Очікування: очікується на запити про підключення клієнта або запити на оновлення політики.

- Запит на підключення клієнта: запускається перехід до «Автентифікації клієнта».

- Запит на оновлення політики: ініціюється перехід до «Оновлення політик безпеки».

4. Автентифікація клієнта: граничний вузол автентифікує клієнта за допомогою імені користувача та пароля.

- Перевірка облікових даних: виконується перевірка облікових даних.

- Успіх автентифікації: якщо автентифікація пройшла успішно, відбувається перехід до «Видати маркер».

- Неуспішна автентифікація: якщо автентифікація не вдалася, відбувається перехід до «Стан помилки».

5. Видача токену: edge вузол генерує та видає токен з невеликим життєвим циклом для аутентифікованих клієнтів

- Генерація токену: токен генерується

- Успішна видача токену: якщо токен видано успішно, стан стає “Готовий”.

- Помилка видачі токену: якщо токен видано з помилкою, стан переходить до "Стан помилки".

6. Моніторинг активності клієнта: граничний вузол безперервно відстежує активність клієнта на предмет підозрілої поведінки або закінчення терміну дії токена.

- Моніторинг: Граничний вузол відстежує поведінку клієнта.

- Термін дії токена: Якщо термін дії токена закінчується, відбувається перехід до «Поновлення токена».

- Виявлена аномалія: Якщо виявлено аномалію, відбувається перехід до «Обробити аномалію».

7. Оновлення маркерів: Граничний вузол поновлює токен для клієнта.

- Оновлення токена: Токен поновлюється.

- Успіх поновлення токена: Якщо оновлення токена пройшло успішно, він переходить в стан «Готовий».

- Неуспішне оновлення токена: Якщо оновлення токену завершилося невдало, він переходить у стан «Помилка».

8. Оновлення політик безпеки: Граничний вузол оновлює політики безпеки на основі нових правил або даних про загрози.

- Оновлення політик: Виконується оновлення політик.

- Успіх оновлення політик: Якщо оновлення пройшло успішно, він переходить у стан «Готовий».

- Неуспішне оновлення політики: Якщо оновлення завершилося невдало, відбувається перехід до стану «Помилка».

9. Обробити аномалію: Граничний вузол виявляє і обробляє аномалії в поведінці клієнта або шаблонах зв'язку.

- Аналізувати аномалію: Аномалія аналізується.

- Аномалія вирішена: Якщо аномалію вирішено, вузол переходить у статус «Готовий».

- Критична аномалія: Якщо аномалія критична, вона переходить у стан «Помилка».

10. Стан помилки: Граничний вузол обробляє критичні помилки або повторні збої.

- Обробка помилки: Помилка обробляється.

- Вирішено помилку: Якщо помилку вирішено, він переходить у стан «Готовий».

- Незмінна помилка: Якщо помилка не зникає, відбувається перехід до «Вимкнення».

11. Вимкнення: Граничний вузол безпечно завершує свою роботу.

- Вимикання: Граничний вузол завершує роботу.

- Кінець: Кінцевий стан після завершення роботи.

Діаграма процесів для граничного вузла

Діаграма процесів (рис. 2.4) для edge вузла деталізує покрокові операції, які він виконує, ілюструючи логічний потік завдань і рішень. Це допомагає зрозуміти послідовні операції та точки прийняття рішень у процесах граничного вузла.

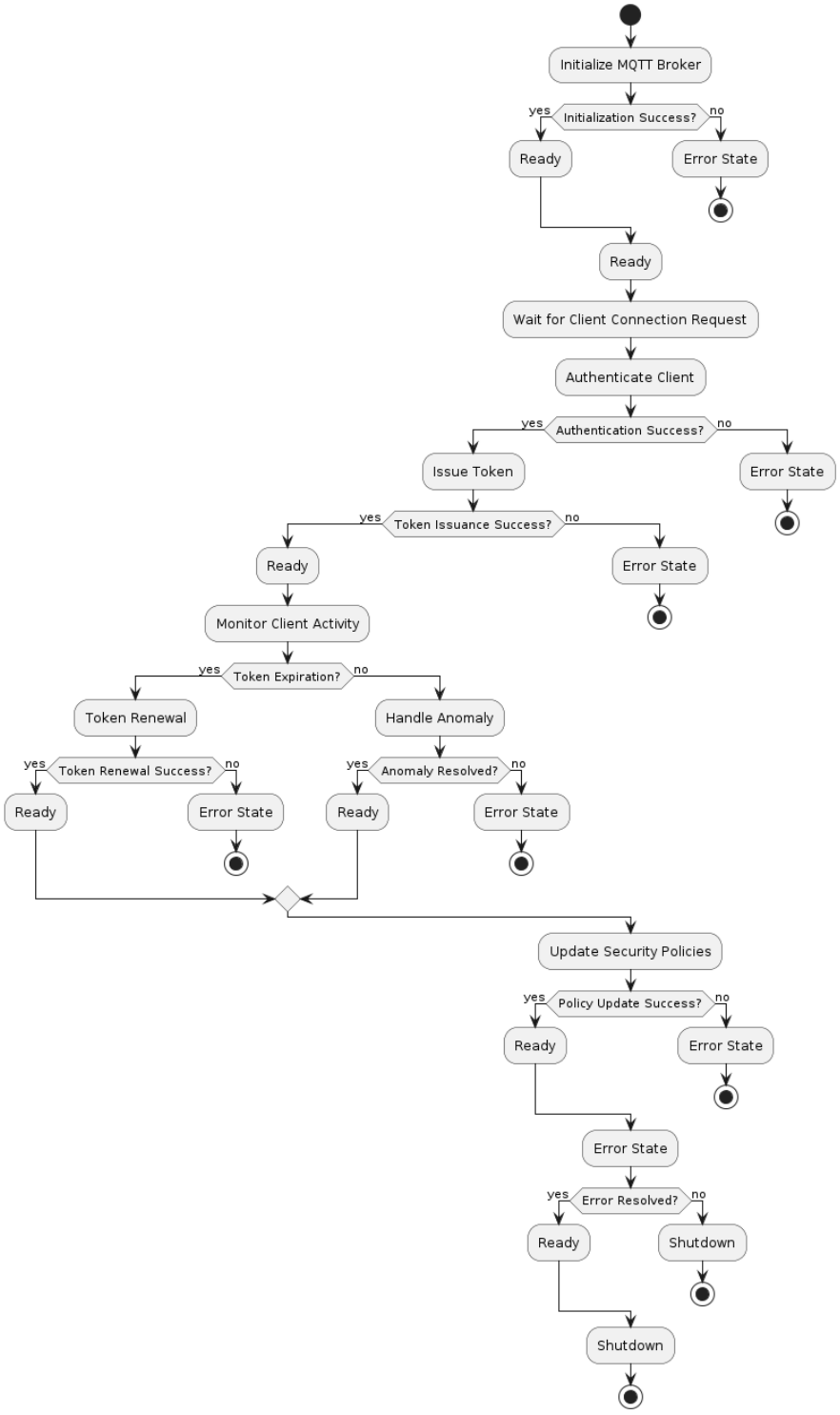

Рисунок 2.4 – Діаграми процесів агента

Опис діаграми

Старт: початкова точка, з якої edge вузол починає свій процес.

Ініціалізувати MQTT-брокер: Граничний вузол ініціалізує MQTT-брокер.

- Успіх ініціалізації: Якщо ініціалізація пройшла успішно, процес переходить у стан «Готовий».

- Неуспішна ініціалізація: Якщо ініціалізація не вдалася, процес переходить у «Стан помилки» і зупиняється.

3. Готовність: Граничний вузол очікує запит на підключення клієнта або запит на оновлення політики.

4. Автентифікація клієнта: Граничний вузол автентифікує клієнта.

- Успіх автентифікації: Якщо автентифікація пройшла успішно, процес переходить до «Видати маркер».

- Неуспішна автентифікація: Якщо автентифікація не вдалася, процес переходить у стан «Помилка» і зупиняється.

5. Видати маркер: Граничний вузол видає токен автентифікованому клієнту.

- Успіх видачі токена: Якщо видача токена пройшла успішно, процес переходить до «Моніторингу активності клієнта».

- Неуспішна видача токенів: Якщо видача токенів не вдалася, процес переходить в «Стан помилки» і зупиняється.

6. Моніторинг активності клієнта: Граничний вузол відстежує активність клієнта.

- Закінчення терміну дії токена: Якщо термін дії токена закінчується, процес переходить до «Поновлення токена».

- Виявлена аномалія: Якщо виявлено аномалію, процес переходить до «Обробки аномалії».

7. Поновлення маркерів: Граничний вузол поновлює токен.

- Успіх оновлення токена: Якщо оновлення токену пройшло успішно, процес повертається до «Моніторингу активності клієнта».

- Неуспішне оновлення токена: Якщо оновлення токену не вдалося, процес переходить в «Стан помилки» і зупиняється.

8. Обробити аномалію: Граничний вузол обробляє виявлені аномалії.

- Вирішено аномалію: Якщо аномалію вирішено, процес повертається до «Моніторингу активності клієнта».

- Критична аномалія: Якщо аномалія критична, процес переходить в «Стан помилки» і зупиняється.

9. Оновлення політик безпеки: Граничний вузол оновлює свої політики безпеки.

- Успіх оновлення політики: Якщо оновлення пройшло успішно, процес повертається до «Моніторингу активності клієнта».

- Неуспішне оновлення політики: Якщо оновлення завершилося невдало, процес переходить у стан «Помилка» і зупиняється.

10. Стан помилки: Граничний вузол обробляє помилки.

- Error Resolved: Якщо помилку вирішено, процес повертається до стану «Готовність»

- Незмінна помилка: Якщо помилка не зникає, процес переходить у стан «Завершення роботи» і зупиняється.

11. Завершення роботи: Граничний вузол завершує свою роботу.

- End: Фінальна точка після завершення роботи.

2.4 Модель безпеки з нульовою довірою

Модель Zero-Trust має важливе значення для забезпечення надійної безпеки в розподілених системах, особливо в edge обчисленнях, де пристрої працюють в різноманітних і часто непередбачуваних умовах. У розділі розглядаються принципи і стратегії реалізації Zero-Trust в рамках запропонованої моделі, підкреслюючи, як вона підвищує безпеку завдяки безперервній перевірці, суворому контролю доступу і виявленню загроз в режимі реального часу.

Взаємна автентифікація

Взаємна автентифікація є основою моделі безпеки Zero-Trust, гарантуючи, що агент і edge вузол постійно перевіряють ідентичність один одного. На відміну від традиційних моделей, які довіряють пристроям після автентифікації, Zero-Trust вимагає постійної перевірки при кожній взаємодії.

Принципи взаємної автентифікації:

Безперервна перевірка: Обидва суб'єкти повинні автентифікувати один одного при кожній спробі зв'язку. Це передбачає початкову автентифікацію за допомогою імені користувача/пароля, за якою слідують механізми на основі токенів для підтримання дійсності сеансу.

Надійні криптографічні методи: Використання стійких криптографічних методів, таких як ChaCha20Poly1305, гарантує безпечну передачу та перевірку автентифікаційних даних.

Динамічні облікові дані: Регулярна зміна токенів та облікових даних мінімізує ризик їх компрометації та використання неавторизованими особами.

Реалізація в запропонованій моделі:

Початкова автентифікація: Коли агент підключається до edge вузла, він використовує безпечну комбінацію імені користувача та пароля. Цей крок гарантує, що агент є легітимним суб'єктом, розпізнаним системою.

Автентифікація на основі токенів: Після первинної автентифікації агент отримує короткочасний токен, який він повинен використовувати в подальших комунікаціях. Граничний вузол перевіряє цей токен щоразу, коли агент надсилає дані або запитує послуги.

Механізм оновлення токенів: Для підтримки безпечного сеансу зв'язку граничний вузол періодично випускає нові токени. Агент повинен запитувати і використовувати ці нові токени протягом їхнього терміну дії, гарантуючи, що облікові дані для автентифікації є динамічними і тимчасовими.

Принцип найменших привілеїв

Даний принцип є важливим для зменшення поверхні атаки та потенційної шкоди у випадку порушення безпеки. Надаючи агенту доступ лише до необхідних йому ресурсів і тем, система мінімізує ризик несанкціонованого доступу або зловмисних дій.

Принцип найменших привілеїв:

- Мінімальні права доступу: агентам надається мінімальний доступ, необхідний для виконання їхніх функцій.
- Контроль доступу на основі ролей (RBAC): доступ визначається на основі ролі агента, забезпечуючи відповідність дозволів операційним потребам.
- Динамічне налаштування: дозволи на доступ можна динамічно змінювати відповідно до змін у вимогах або виявлених загроз.

Реалізація в запропонованій моделі:

- Контроль доступу на рівні теми: edge вузол обмежує доступ агента Rust до певних тем MQTT. Це гарантує, що агент може публікувати або підписуватися лише на теми, необхідні для його роботи.
- Визначення ролей: система визначає ролі та призначає дозволи на основі цих ролей. Наприклад, агент моніторингу може мати інші права доступу, ніж агент контролю.
- Періодичні перегляди: Права доступу періодично переглядаються і коригуються на основі моделей використання і нових потреб безпеки. Такий проактивний підхід допомагає виявити та відкликати непотрібні привілеї.

Безперервний моніторинг

Безперервний моніторинг - це проактивна стратегія виявлення та реагування на загрози в режимі реального часу. Він передбачає постійне спостереження за мережевою діяльністю, потоками даних і станом системи для виявлення аномалій або підозрілої поведінки, які можуть свідчити про порушення безпеки.

Принципи безперервного моніторингу:

Аналіз у режимі реального часу: система безперервно аналізує дані та мережеву активність, щоб виявити відхилення від нормальних шаблонів.

Автоматичні сповіщення та реагування: автоматизовані системи генерують сповіщення та запускають заздалегідь визначені реакції на виявлені загрози, мінімізуючи час реагування.

Комплексне ведення журналів: ведеться детальний облік усіх дій.

Реалізація в запропонованій моделі:

Виявлення аномалій: граничний вузол відстежує поведінку агента, шукаючи незвичайні патерни в передачі даних, використанні токенів і запитах на доступ.

Наприклад, несподіваний сплеск частоти повідомлень або повторювані збої автентифікації можуть викликати сповіщення.

Автоматичне реагування: при виявленні аномалії, система може вжити автоматичних заходів, таких як ізоляція підозрілого агента, відкликання його доступу або ескалація проблеми до людей-операторів.

Ведення журналів: Всі взаємодії та події повністю реєструються в журналах. Ці журнали надійно зберігаються і можуть бути використані для аналізу після інциденту з метою вдосконалення заходів і політик безпеки.

Додаткові заходи безпеки

Окрім основних принципів взаємної автентифікації, найменших привілеїв та постійного моніторингу, запропонована модель нульової довіри включає додаткові заходи безпеки для подальшого підвищення надійності системи.

Сегментація мережі: мережа поділяється на сегменти на основі рівнів довіри та функціональних вимог. Така сегментація допомагає стримувати потенційні порушення, обмежуючи поширення атаки в мережі.

Пісочниця: певні операції, особливо ті, що передбачають обробку ненадійних даних або виконання неперевіреного коду, виконуються в ізольованих середовищах (пісочницях). Така ізоляція гарантує, що будь-які зловмисні дії не вплинуть на основну систему.

Багатофакторна автентифікація (MFA): для особливо важливих операцій або адміністративного доступу використовується MFA. Це додає додатковий рівень безпеки, вимагаючи додаткових кроків перевірки, окрім паролів або токенів.

Інтеграція thread intelligence: система інтегрується із зовнішніми джерелами розвідки загроз, щоб бути в курсі нових загроз. Ця інформація використовується для динамічного коригування політик безпеки та реагування на загрози.

Регулярний аудит безпеки та тестування на проникнення: система проходить регулярний аудит безпеки та тестування на проникнення для виявлення

вразливостей і забезпечення відповідності стандартам безпеки. Ці проактивні заходи допомагають підтримувати високий рівень безпеки.

Прийнявши модель безпеки Zero-Trust, запропонована система гарантує, що кожен суб'єкт, як внутрішній, так і зовнішній, постійно перевіряється, і йому надається лише мінімально необхідний доступ. Постійний моніторинг та додаткові заходи безпеки ще більше підвищують здатність системи виявляти загрози та реагувати на них у режимі реального часу. Такий комплексний підхід до безпеки має важливе значення для збереження цілісності, конфіденційності та доступності ресурсів в edge-орієнтованому середовищі.

2.5 Оцінка ризиків

Оцінка ризиків є критично важливим процесом в управлінні безпекою для периферійних архітектур. Він включає в себе виявлення потенційних ризиків, оцінку їх ймовірності та впливу, а також розробку стратегій пом'якшення для зменшення загального ризику для системи. NIST RMF пропонує комплексний підхід, який інтегрує безпеку, конфіденційність та управління ризиками в життєвий цикл розробки системи.

Цілі оцінки ризиків:

- Ідентифікація ризиків: Визначити потенційні загрози безпеці, які можуть вплинути на систему.
- Оцінка ризиків: Оцінити ймовірність та вплив виявлених ризиків.
- Визначення пріоритетності ризиків: Ранжування ризиків за ступенем їхньої серйозності, щоб зосередитися на найбільш критичних з них.
- Пом'якшення ризиків: Розробка стратегій для зменшення або усунення виявлених ризиків.

Методологія оцінки ризиків

NIST RMF[9] складається з семи кроків, які забезпечують системний підхід до управління ризиками:

1. Підготовка:
    1. Мета: забезпечення готовності організації до управління ризиками безпеки та конфіденційності.
    2. Заходи: визначення ролі в управлінні ризиками, розробка стратегії управління ризиками та встановлення рівня толерантності до ризиків.
2. Категоризація:
    1. Мета: категоризація інформаційної системи та інформації, що обробляється, зберігається та передається, на основі аналізу впливу.

2. Заходи: присвоєння категорії безпеки компонентам системи на основі їхнього потенційного впливу на організацію.

3. Вибір:
    1. Мета: обрання відповідних засобів контролю безпеки для системи на основі оцінки ризиків.
    2. Заходи: визначення та адаптування засобів контролю безпеки з NIST SP 800-53, враховуючи конкретні потреби та контекст системи.

4. Впровадження:
    1. Мета: впровадження вибраних засобів контролю безпеки та документування їх розгортання.
    2. Заходи: впровадження засобів контролю безпеки та забезпечення їх інтеграцію в архітектуру та процеси системи.

5. Оцінка:
    1. Мета: оцінка ефективності засобів контролю безпеки, щоб переконатися, що вони працюють належним чином.
    2. Заходи: проведення оцінювання засобів контролю безпеки, документування результатів та усунення будь-яких виявлених недоліків.

6. Надання повноважень:
    1. Мета: надання дозволу на використання системи з урахуванням ризику для діяльності організації, активів та окремих осіб.
    2. Заходи: підготування пакету дозвільних документів, оцінка ризиків та прийняття рішення про надання дозволу на використання системи на основі оцінки ризиків.

7. Моніторинг:

1. Мета: Постійний моніторинг засобів контролю безпеки та стану безпеки системи для забезпечення постійної ефективності.
2. Заходи: Впровадження стратегії безперервного моніторингу, проведення регулярних оцінок та реагування на будь-які зміни або інциденти.

Виявлення потенційних ризиків передбачає вивчення компонентів системи, потоків даних та операційного середовища для виявлення вразливостей і загроз. Цей крок має вирішальне значення для розуміння того, де і як система може бути скомпрометована.

Оцінка ризиків

Оцінка ризиків оцінює ймовірність і вплив виявлених ризиків, допомагаючи визначити їх пріоритетність на основі їх потенційної серйозності.

Оцінка ймовірності:

- Низька ймовірність: Рідкісні випадки, мінімальний вплив або вразливості, які важко використати.
- Середня ймовірність: Можливі випадки з деяким впливом або помірною легкістю використання.
- Висока ймовірність: Часті випадки, високий рівень впливу або вразливості, які легко використати.

Оцінка впливу:

- Низький рівень впливу: Мінімальні операційні збої, низькі фінансові втрати або відсутність значної компрометації даних.
- Середній вплив: Помірні перебої в роботі, керовані фінансові втрати або деяка компрометація даних.
- Високий рівень впливу: Значні перебої в роботі, великі фінансові втрати або серйозна компрометація даних

Враховуючи описані в 1 розділі ризики для edge-орієнтованих інфраструктур, матриця ризиків має наступний вигляд (табл. 2.1):

Таблиця 2.1 — Матриця ризиків для edge-орієнтованих інфраструктур

Ризик	Ймовірність	Вплив	Пріоритет
Вразливості на фізичному рівні/Апаратні вразливості	Середня	Високий	Високий
Вразливості програмного забезпечення	Висока	Висока	Висока
Мережеві атаки	Середня	Середній	Середній
Проблеми з автентифікацією	Висока	Висока	Висока
Проблеми з конфіденційністю даних	Середня	Висока	Висока

1. **Апаратні вразливості**:

Ймовірність: Середня. Фізичний доступ до периферійних пристроїв можна обмежити, цілеспрямовані зловмисники все одно знайдуть способи використати апаратні вразливості.

Вплив: високий. Апаратні вразливості можуть призвести до серйозних наслідків, таких як несправність пристрою, несанкціонований доступ або незворотні пошкодження.

2. **Вразливості програмного забезпечення**:

Імовірність: висока. Вразливості в програмному забезпеченні є поширеним явищем і можуть виникати як через помилки при проектуванні, та і через помилки при реалізації, застарілі бібліотеки або неправильну конфігурацію тощо.

Вплив: високий. Використання вразливостей програмного забезпечення може призвести до несанкціонованого доступу, витоку даних та компрометації системи.

3. **Мережеві загрози**:

Ймовірність: Середня. Мережеві загрози, такі як атаки типу «man-in-the-middle» або DoS-атаки, є поширеними, але їх можна зменшити за допомогою належних заходів мережевої безпеки.

Вплив: Середній. Мережеві загрози можуть призвести до втрати цілісності чи доступності.

4. **Проблеми з аутентифікацією**:

Ймовірність: Висока. Механізми автентифікації часто стають мішенню зловмисників.

Вплив: високий. Порушення автентифікації може призвести до несанкціонованого доступу до конфіденційних даних та критично важливих систем.

5. **Питання конфіденційності даних**:

Ймовірність: Середня. Ризики для конфіденційності даних виникають через неналежний контроль доступу, витоки даних або ненадійне шифрування.

Вплив: високий. Порушення конфіденційності даних може призвести до значної юридичної, фінансової та репутаційної шкоди.

Кількісна оцінка ризиків

Використаємо кількісну оцінку ризиків, на основі раніше складеної матриці ризиків, що передбачає розрахунок ризику для кожної ідентифікованої загрози на основі її ймовірності та впливу[11, 12, 13].

Для оцінки ризику можна використати наступну формулу (2.1):

$$Risk = Likelihood * Impact \quad (2.1)$$

, де:

- Risk — обраховуваний ризик
- Likehood (L) — ймовірність реалізації загрози протягом певного періоду часу, як правило, виражається десятковим дробом від 0 до 1.
- Impact (I) -- це потенційна шкода або втрата, пов'язана із загрозою, яка може бути виражена в грошовому еквіваленті або інших відповідних одиницях[14,15].

Оцінка ймовірності

Оцінка ймовірності обраховується за допомогою наступної формули (2.2):

$$L = P(\text{виникнення загрози}) \quad (2.2)$$

Оцінка впливу

Оцінка впливу кількісно визначає потенційні наслідки кожного ризику. Це може включати фінансові втрати, серйозність витоку даних, операційні перебої та інші відповідні фактори. Вплив виражається у грошовому еквіваленті або у вигляді комплексної оцінки за формулою (2.3):

$$I = \text{Фін. збитки} + \text{Серйозн. витокуданих} + \text{Перебоївроботі} \quad (2.3)$$

Використаємо формули(2.1 та 2.2) для обрахунку кожного ризику з ризиків, названих раніше, взявши за суму впливу певне число N, зробивши формулу більш універсальною:

Апаратні вразливості:

Ймовірність: L=0.3

Вплив: I=N

Ризик: $Risk = 0.3 * N = 0.3N$

Вразливості програмного забезпечення:

Імовірність: L=0.5

Вплив: I=1.5N

Ризик: $Risk = 0.5 * 1.5N = 0.75N$

Мережеві загрози:

Ймовірність: L=0.4

Вплив: I=0.8N

Ризик: $Risk = 0.4 * 0.8N = 0.32N$

Проблеми з автентифікацією:

Ймовірність: L=0.6

Вплив: I=1.2N

Risk: $Risk = 0.6 * 1.2N = 0.72N$

Проблеми з конфіденційністю даних:

Ймовірність: L=0.4

Вплив: I=1.8N

Ризик: $Risk = 0.4 * 1.8N = 0.72N$

Для отримання загального ризику, підсумуємо всі ризики за формулою (2.4):

$$TotalRisk = \sum(Risk_i) \quad (2.4)$$

Отже, підрахувавши, маємо загальний ризик:

$$TotalRisk = 0.3N + 0.75N + 0.32N + 0.72N + 0.72N = 2.81N$$

Застосування агентної моделі

Запропонована агентна модель використовує різні механізми для усунення виявлених ризиків:

1. **Апаратні вразливості**:

- Стратегія мінімізації: Модель включає механізми безпечного завантаження(secure boot) та використовує апаратні модулі безпеки (HSM) для захисту криптографічних ключів та операцій.
- Переваги: Підвищує фізичну безпеку, запобігаючи несанкціонованому втручанню та забезпечуючи цілісність криптографічних операцій.
- Недоліки: Впровадження HSM може бути дорогим і вимагати додаткових апаратних ресурсів, що може бути обмеженням у середовищах з обмеженими ресурсами. Також використання безпечного завантаження є важким в реалізації, тому даний тим захисту і є опціональним.

Нова ймовірність ризику:

- Ймовірність: L=0.2
- Новий ризик: $NewRisk = 0.2 * N = 0.2N$

2. **Вразливості програмного забезпечення**:

- Стратегія мінімізації: при написанні Агенту та Edge вузла, має дотримуватися безпечних практик програмування, проходячи регулярні перевірки коду та за можливості, отримання оновлень по повітрю (OTA) для виправлень та вдосконалень мір безпеки.
- Переваги: Регулярні оновлення та безпечне програмування значно знижують ризик використання вразливостей у програмному забезпеченні.
- Недоліки: оновлення OTA вимагають надійного мережевого підключення, яке не завжди може бути доступним у віддалених периферійних середовищах.

Нова ймовірність ризику:

- Ймовірність: L=0.3
- Новий ризик: $NewRisk = 0.2 * 1.5N = 0.45N$

3. **Мережеві загрози**:

- Стратегія мінімізації: Модель використовує надійні протоколи шифрування, такі як ChaCha20Poly1305, для даних, що передаються, і застосовує сегментацію мережі для ізоляції критично важливих компонентів.

- Переваги: Надійне шифрування та сегментація мережі захищають цілісність і конфіденційність даних, знижуючи ризик мережевих атак.
- Недоліки: шифрування і сегментація можуть призвести до затримок і ускладнення управління мережею.

Нова ймовірність ризику:

- Ймовірність: L=0.2
- Новий ризик: $NewRisk = 0.2 * 0.8N = 0.16N$

4. **Питання автентифікації та авторизації**:

Стратегія мінімізації: Модель реалізує багатофакторну автентифікацію (MFA) для критично важливих операцій, регулярно проводить ротацію токенів автентифікації та впроваджує політики контролю доступу з найменшими привілеями.

Переваги: MFA та ротація токенів забезпечують надійні механізми автентифікації, значно знижуючи ризик несанкціонованого доступу.

Недоліки: Впровадження та управління MFA може бути складним і вимагати додаткового навчання та підтримки користувачів.

Нова ймовірність ризику:

- Ймовірність: L=0.3
- Новий ризик: $NewRisk = 0.3 * 1.2 * N = 0.36N$

5. **Конфіденційність та цілісність даних**:

Стратегія мінімізації: Модель використовує наскрізне шифрування даних у стані спокою та під час передачі, а також впроваджує механізми перевірки та моніторингу цілісності даних.

Переваги: Забезпечує конфіденційність і недоторканність даних протягом усього їхнього життєвого циклу, знижуючи ризики конфіденційності та цілісності.

Недоліки: шифрування та перевірка цілісності можуть споживати обчислювальні ресурси, що потенційно впливає на продуктивність пристроїв.

Нова ймовірність ризику:

- Ймовірність: L=0.2
- Новий ризик: $NewRisk = 0.2 * 1.8N = 0.36N$

Складемо таблицю(2.2) з усіма ризиками що були обраховані, та обчислимо зменшення ризиків завдяки методам та підходам що вказані в запропонованій агентній моделі:

Таблиця 2.2 — Таблиця обрахованих ризиків

Тип ризику	Оригінальний ризик	Новий ризик	Зменшення ризику
Вразливості на фізичному рівні/Апаратні вразливості	0.3N	0.2N	0.1N
Вразливості програмного забезпечення	0.75N	0.45N	0.3N
Мережеві атаки	0.32N	0.16N	0.16N
Проблеми з автентифікацією	0.72N	0.36N	0.36N
Проблеми з конфіденційністю даних	0.72N	0.36N	0.36N

Обрахуємо зменшення ризику відносно початкового за формулою(2.5):

$$RiskReduction = {\sum(Risk_i)}\Big/{\sum(NewRisk_i)} \quad (2.5)$$

Отримуємо, що зменшення ризику дорівнює:

$$NewRisk = \sum_i NewRisk_i = 1.53N$$

$$RiskReduction = 2.81N \ / \ 1.53N \approx 1.84$$

Висновки за розділом 2

У даному розділі описано комплексну модель безпеки для Edge-орієнтованих архітектур з використанням агентного підходу; детально описується архітектура, яка включає агент і Edge вузол, що функціонує як брокер MQTT і менеджер безпеки. Агент забезпечує безпечний зв'язок за допомогою протоколів MQTT і шифрування ChaCha20Poly1305, в той час як периферійний вузол займається генерацією токенів, аутентифікацією і безперервним моніторингом загроз.

NIST Risk Management Framework (RMF) застосовується для систематичного виявлення, оцінки та пом'якшення потенційних ризиків безпеки. Ключові ризики включають вразливість апаратного та програмного забезпечення, мережеві загрози, проблеми з автентифікацією та авторизацією, а також проблеми з конфіденційністю та цілісністю даних. Матриця ризиків використовується для оцінки цих ризиків на основі їхньої ймовірності та впливу, що допомагає визначити пріоритети стратегій пом'якшення наслідків. Модель на основі агентів усуває ці ризики за допомогою механізмів безпечного завантаження, апаратних модулів безпеки, безпечного проектування та програмування коду, надійного шифрування, багатофакторної автентифікації та безперервного моніторингу. У розділі висвітлюються переваги використання токенів для безперервної автентифікації та безпечної передачі даних, а також визнаються складнощі та витрати ресурсів, пов'язані з цим. Застосовуючи запропоновані методи можна зменшити загальні ризики, пов’язані з найпоширенішими атаками приблизно в 1.5-2 рази.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Тест продуктивності

В якості демонстрації, пропонована модель була реалізована мовою Rust. Моделювання проводилися на віртуальній машині з обмеженою кількістю ресурсів:

- 2 Гб оперативної пам’яті
- 1 ядро AMD Ryzen 5 4500U
- 20 Гб місяця на диску

Продуктивність моделі вимірювалася шляхом запису використання процесора та пам'яті з різною кількістю клієнтів. Протестовані наступні клієнтські навантаження зазначені у табл. 3.1:

Таблиця 3.1 — Залежність ресурсів від кількості кінцевих пристроїв

Кількість клієнтів	Використання пам'яті	Навантаження на процесор
5	0.8 kB	0%
10	2.7 Mb	0%
25	3.1 Mb	0.1%
50	3.4 Mb	93%
Агент	732 kB	0%

Для кожного тесту відстежувалося і записувалося використання процесора і пам'яті протягом певного періоду часу. З цих даних можна побудувати графік залежності кількості ресурсів що необхідні для роботи, від кількості кінцевих пристроїв

Нижче побудовано графік залежності навантаженості на процесор від кількості клієнтів (рис. 3.1):

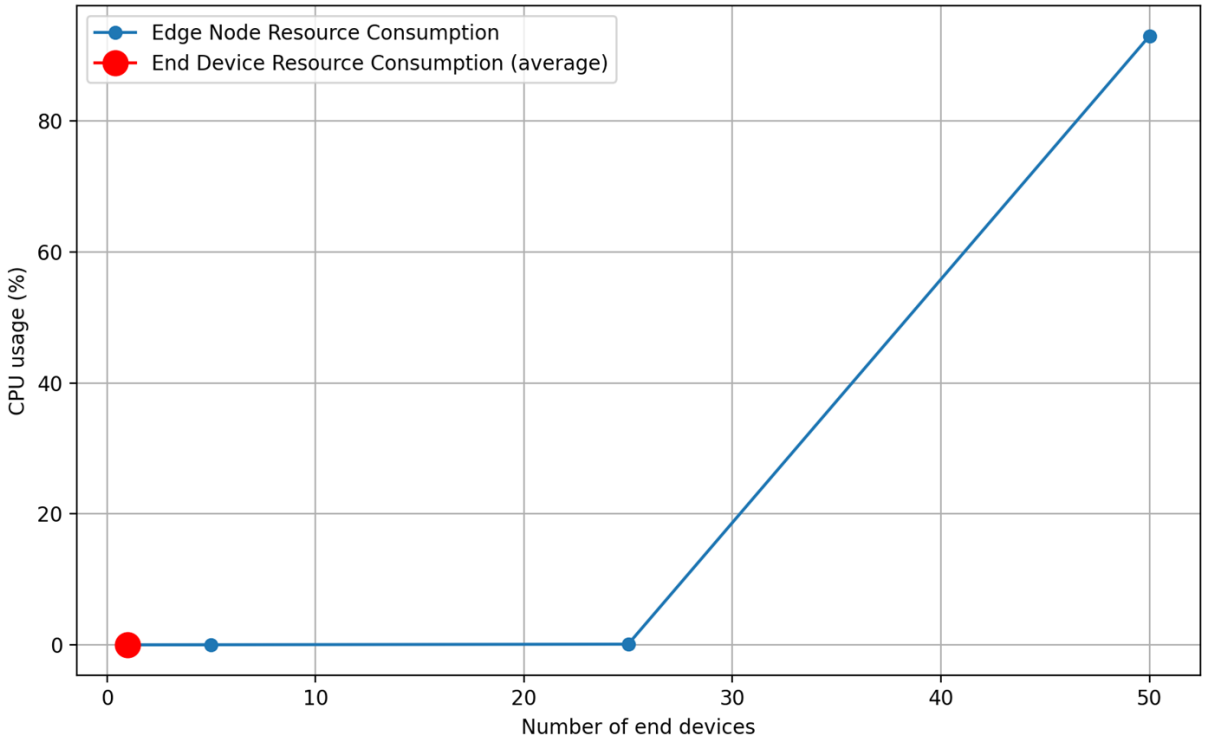

Рисунок 3.1 -- Графік залежності навантаженості на процесор від кількості клієнтів

, а також графік залежності завантаженості пам'яті від кількості клієнтів (рис. 3.2)

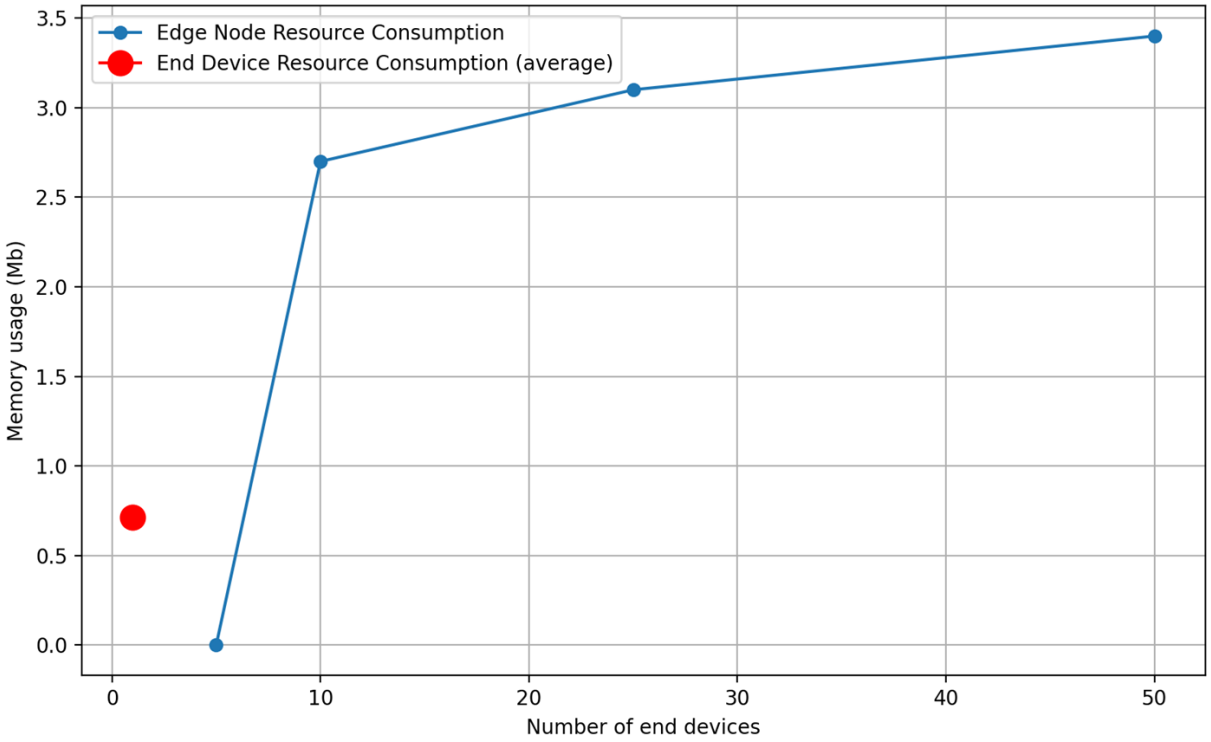

Рисунок 3.2 -- Графік залежності використання пам’яті від кількості клієнтів

3.2 Моделювання атак

Для підтвердження, що зв'язок між клієнтами та Edge вузлом є безпечним, було використано Wireshark для перехоплення мережевого трафіку. Захоплений трафік було проаналізовано, та перевірено чи всі MQTT-повідомлення були зашифровані належним чином(рис 3.3).

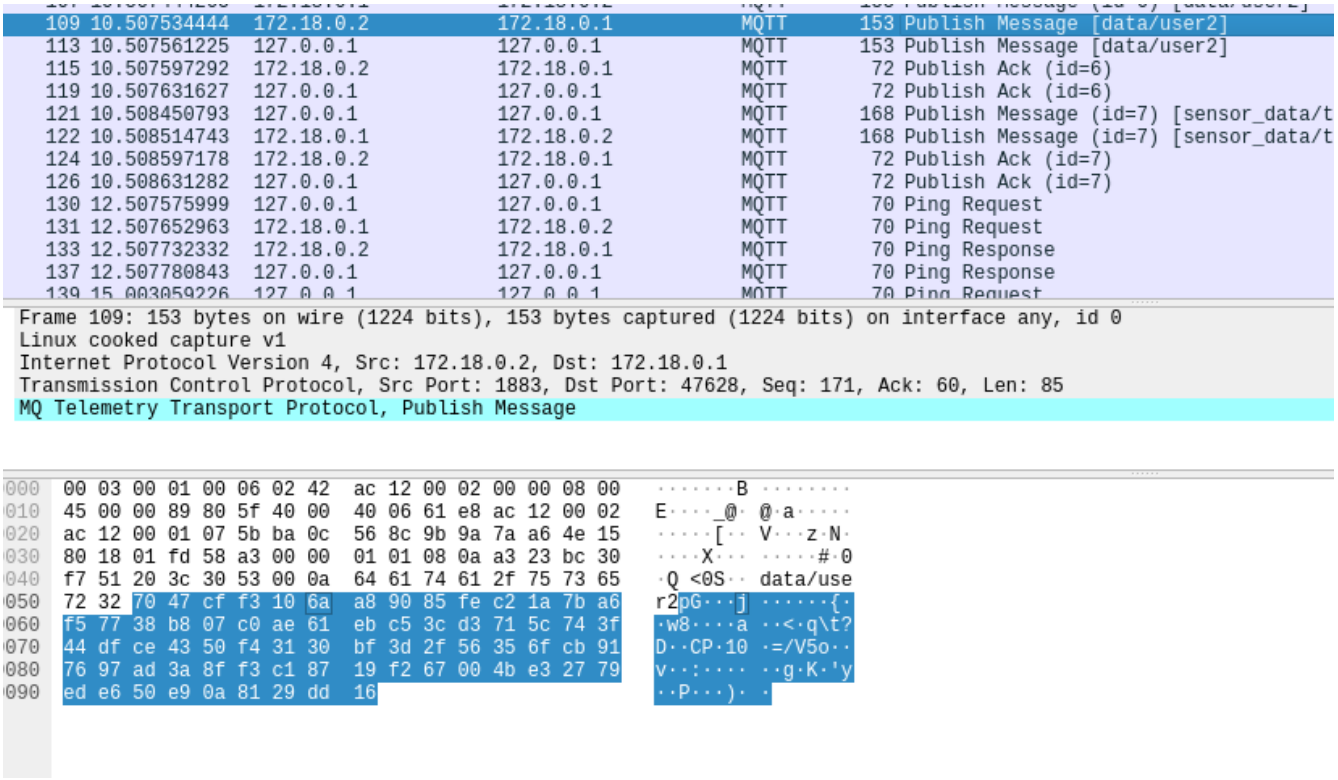

Рисунок 3.3 — Перевірка шифрування трафіку

Також було проведено експеримент для перевірки реакції системи на недійсні токени. Клієнт спробував пройти аутентифікацію за допомогою недійсного токену, і були зроблені наступні спостереження:

- граничний вузол виявив недійсний токен.

- клієнт був виключений зі списку спостережуваних.

Демонстрація логів серверу на рис 3.4:

```
→  srv git:(master) ✗ cat logs/user2.log
LOG: handling user2     [2024-06-07][20-06-1717780062]
LOG: current token: "02d7VuD7GO7gHiuUsomDyTbIDhQqnX36"
old token: "invalid"    [2024-06-07][20-06-1717780062]
LOG: processing data    [2024-06-07][20-06-1717780066]
ERROR: token was not validated  [2024-06-07][20-06-1717780066]
LOG: processing data    [2024-06-07][20-06-1717780069]
ERROR: token was not validated  [2024-06-07][20-06-1717780069]
LOG: processing data    [2024-06-07][20-06-1717780069]
ERROR: token was not validated  [2024-06-07][20-06-1717780069]
LOG: processing data    [2024-06-07][20-06-1717780070]
ERROR: token was not validated  [2024-06-07][20-06-1717780070]
LOG: processing data    [2024-06-07][20-06-1717780071]
ERROR: token was not validated  [2024-06-07][20-06-1717780071]
LOG: processing data    [2024-06-07][20-06-1717780071]
ERROR: token was not validated  [2024-06-07][20-06-1717780071]
LOG: processing data    [2024-06-07][20-06-1717780072]
ERROR: token was not validated  [2024-06-07][20-06-1717780072]
LOG: processing data    [2024-06-07][20-06-1717780073]
ERROR: token was not validated  [2024-06-07][20-06-1717780073]
LOG: processing data    [2024-06-07][20-06-1717780073]
ERROR: token was not validated  [2024-06-07][20-06-1717780073]
LOG: processing data    [2024-06-07][20-06-1717780074]
ERROR: token was not validated  [2024-06-07][20-06-1717780074]
WARNING: exceeded maximum token attempts, terminating...        [2024-06-07][20-06-1717780074]
→  srv git:(master) ✗
```

Рисунок 3.4 — Реакція Edge вузла на перевищення ліміту помилкових токенів

Висновки за розділом 3

Експерименти продемонстрували реалістичність та ефективність запропонованої моделі безпеки для гранично-орієнтованих архітектур. Основні висновки включають

Модель ефективно обробляла різну кількість клієнтів з керованим використанням процесора та пам'яті.

Зашифрований зв'язок був успішно перевірений, що забезпечило конфіденційність даних.

Система ефективно обробляла недійсні токени, підвищуючи безпеку завдяки безперервній автентифікації та застосуванню.

Систематично оцінюючи продуктивність і безпеку запропонованої моделі, ці експерименти підтверджують її застосовність у реальних середовищах периферійних обчислень, забезпечуючи безпечну та ефективну роботу.

4 РОЗРОБКА СТАРТАП ПРОЕКТУ

4.1 Опис ідеї проекту

Проект спрямований на розробку комплексного рішення безпеки, спеціально розробленого для Edge-орієнтовних архітектур, з урахуванням унікальних ризиків безпеки, притаманних розподіленій природі периферійних обчислень.

Запропоноване рішення фокусується на захисті даних ближче до їх джерела, що має вирішальне значення в різних сферах - від промислової автоматизації до охорони здоров'я та розумних міст. Впроваджуючи модель безпеки на основі агентів, цей проект має намір підвищити конфіденційність, надійність і швидкість обробки даних на периферії.

Запропонована модель використовуватиме протоколи MQTT і шифрування ChaCha20Poly1305 для безпечного зв'язку, а периферійні вузли будуть виступати в ролі брокерів MQTT і менеджерів безпеки для забезпечення надійного моніторингу та управління загрозами

На відміну від існуючих аналогів, інноваційність проекту полягає в індивідуальному підході до безпеки для різних галузей, що інтегрує NIST Risk Management Framework (RMF) для систематичного управління ризиками. Рішення не лише захищає дані, але й безперервно відстежує та реагує на загрози в режимі реального часу, забезпечуючи значну перевагу над традиційними централізованими моделями безпеки.

4.2 Технологічний аудит ідеї проекту

Технологічна основа цього проекту базується на перевірених часом і широко доступних технологіях, таких як протоколи MQTT для черги повідомлень, ChaCha20Poly1305 для шифрування та NIST RMF для управління ризиками. Ці технології будуть інтегровані для створення надійної моделі безпеки edge-орієнтованих систем. Здійсненність проекту підтверджується наявністю та доступністю цих технологій для команди розробників. Існуючі технології будуть використані та додатково налаштовані для задоволення специфічних потреб безпеки периферійних обчислювальних середовищ.

Висновок технологічного аудиту є позитивним, підтверджуючи, що реалізація можлива з використанням наявних на даний момент технологій. Команда розробників має необхідний досвід для ефективної інтеграції цих технологій, забезпечуючи практичне та інноваційне рішення безпеки для периферійних обчислень.

4.3 Аналіз ринкових можливостей для запуску стартап-проекту

Ринок рішень для захисту периферійних обчислень стрімко розвивається завдяки стрімкому зростанню пристроїв Інтернету речей та зростаючій потребі в обробці даних у режимі реального часу. Такі галузі, як охорона здоров'я, розумні міста та автономні транспортні засоби, є особливо перспективними, оскільки вони потребують надійних рішень для захисту конфіденційних даних та забезпечення операційної цілісності. Регуляторні вимоги до захисту даних і все більш широке впровадження практик четвертої промислової революції[21] ще більше розширюють ринкові можливості.

Однак ринок також несе в собі загрози, такі як інтенсивна конкуренція з боку відомих фірм, що займаються кібербезпекою, і потенційний опір впровадженню нових технологій у традиційних галузях промисловості. Обмеженість ресурсів, включаючи високі початкові інвестиції та потребу в спеціалізованій експертизі, також є значними викликами. Незважаючи на ці загрози, динаміка ринку є сприятливою для впровадження інноваційного та надійного рішення безпеки, адаптованого для периферійних обчислень.

4.4 Розробка ринкової стратегії проекту

Ринкова стратегія проекту передбачає зосередження на швидкозростаючих ринках, таких як охорона здоров'я, розумні міста та автономні транспортні засоби, де потреба в безпечних периферійних обчисленнях є критично важливою. Партнерство з виробниками пристроїв Інтернету речей та компаніями, що займаються промисловою автоматизацією, матиме важливе значення для завоювання ринку та забезпечення широкого впровадження.

Основна стратегія розвитку зосереджена на інноваційному лідерстві та клієнтоорієнтованості. Постійно впроваджуючи інновації та адаптуючи рішення безпеки до конкретних потреб різних галузей, проект прагне диференціювати себе від конкурентів. Стратегія конкурентної поведінки підкреслюватиме унікальні аспекти моделі безпеки, зокрема, її передові методи шифрування та комплексні можливості моніторингу загроз. Позиціонування стартапу як лідера в галузі безпеки периферійних обчислень буде досягнуто шляхом підкреслення його інноваційності, надійності та здатності відповідати суворим регуляторним вимогам.

4.5 Розробка маркетингової програми

Дана програма зосереджена на трирівневій маркетинговій моделі, комплексній ціновій стратегії, оптимальній системі продажів і добре структурованому плані маркетингових комунікацій.

Трирівнева модель маркетингу:

- Основний продукт: Основним продуктом стартапу є комплексне рішення безпеки для периферійних обчислень. Це рішення включає вдосконалені протоколи шифрування, агентні моделі безпеки та системи безперервного моніторингу загроз. Основний продукт задовольняє фундаментальну потребу в безпечній, надійній та ефективній обробці даних ближче до їхнього джерела, що має вирішальне значення для таких галузей, як охорона здоров'я, розумні міста, промислова автоматизація та автономні транспортні засоби.
- Реальний продукт: Фактичним продуктом є повністю інтегрована система безпеки, яка охоплює апаратні та програмні компоненти. Ця система включає в себе периферійні вузли, що діють як брокери MQTT і менеджери безпеки, оснащені шифруванням ChaCha20Poly1305 для безпечних комунікацій. Фактичний продукт розроблений таким чином, щоб бути масштабованим і адаптованим, що дозволяє його налаштовувати відповідно до конкретних вимог різних галузей і випадків використання.
- Розширений продукт: Розширений продукт складається з послуг підтримки та обслуговування, навчальних програм і безперервних оновлень, які гарантують, що система безпеки залишається стійкою до нових загроз. Сюди входять регулярні оновлення програмного забезпечення, Thread Intelligence у режимі реального часу та послуги підтримки клієнтів, які допомагають їм оптимізувати використання системи безпеки. Надання цих додаткових послуг підвищує рівень задоволеності та лояльності клієнтів, сприяючи розвитку довгострокових відносин з ними.

Цінова стратегія:

Розглянуто кілька моделей ціноутворення:

Ціноутворення на основі передплати: модель передбачає стягнення з клієнтів регулярної плати (щомісячної або щорічної) за доступ до рішення для захисту та постійну підтримку.

Одноразова покупка з контрактами на обслуговування: Клієнти можуть вибрати одноразову покупку системи безпеки з можливістю підписання щорічних контрактів на обслуговування.

Ціноутворення на основі використання: Для клієнтів зі змінними потребами в безпеці може бути реалізована модель ціноутворення на основі використання, яка передбачає оплату на основі обсягу оброблених даних або кількості захищених периферійних пристроїв.

Оптимальна система продажів:

Система продажів буде розроблена таким чином, щоб ефективно охоплювати різні сегменти клієнтів:

- Прямі продажі: Спеціальна команда продажів зосередиться на прямих продажах великим підприємствам, державним установам та постачальникам критично важливої інфраструктури.
- Партнерства та торгові посередники: Налагодження партнерських відносин з виробниками пристроїв Інтернету речей, системними інтеграторами та компаніями, що займаються кібербезпекою, може розширити сферу застосування продукту.

План маркетингових комунікацій:

План маркетингових комунікацій буде спрямований на підвищення обізнаності, генерування інтересу за допомогою різних каналів:

- Контент-маркетинг: Буде створено високоякісний контент, такий як технічні документи, тематичні дослідження та публікації в блогах, щоб розповісти потенційним клієнтам про важливість безпеки периферійних обчислень та унікальні переваги рішення стартапу.
- Пошукова оптимізація (SEO): Впровадження SEO-стратегій допоможе покращити видимість сайту компанії в пошукових системах, залучаючи органічний трафік від потенційних клієнтів, які шукають рішення для захисту периферійних обчислень.
- Вебінари та семінари
- Email-маркетинг
- Зв'язки з громадськістю (PR)
- Виставки та галузеві конференції

Висновки за розділом 4

Комплексний аналіз вказує на високу ймовірність ринкової комерціалізації проекту. Існує значний попит, зумовлений поширенням пристроїв Інтернету речей та суворими регуляторними вимогами щодо захисту даних. Динаміка ринку є сприятливою, з високим потенціалом зростання в таких галузях, як охорона здоров'я, розумні міста та автономні транспортні засоби.

Перспективи впровадження є багатообіцяючими, а потенційні групи клієнтів включають фірми, що займаються промисловою автоматизацією, постачальників медичних послуг та проекти «розумних міст». Незважаючи на бар'єри для входу на ринок, такі як високі початкові витрати і опір новим технологіям, конкурентне середовище дозволяє диференціюватися за рахунок інновацій та надійності. Найкращою альтернативою впровадження є модель прямих продажів для великих підприємств і модель партнерства з виробниками пристроїв Інтернету речей для комплексних пропозицій.

Подальша реалізація цього проекту є доцільною з огляду на високий попит і готовність ринку до передових рішень безпеки в області периферійних обчислень. Унікальний підхід, що поєднує агентні моделі безпеки з просунутим шифруванням і комплексним моніторингом загроз, позиціонує стартап для успіху в мінливому ландшафті кібербезпеки.

ВИСНОВКИ

Дана дипломна робота досліджується розвиток edge-орієнтовних обчислень з акцентом на проблеми безпеки та стратегії управління ризиками, що є важливими для надійного та безпечного розгортання та використання архітектури.

У першому розділі було розглянуто переваги та проблеми граничних обчислень. Edge обчислення обробляють дані ближче до джерела, пропонуючи зменшену затримку, підвищену конфіденційність і поліпшену надійність. У розділі також описані такі фактори, як поширення пристроїв Інтернету речей і попит на аналітику в реальному часі. Області застосування варіюються від промислової автоматизації до «розумних» міст і охорони здоров'я. Незважаючи на ці переваги, граничні обчислення створюють проблеми з безпекою через розподілений і обмежений характер периферійних пристроїв, що робить їх вразливими до різних загроз.

У другому розділі було представлено комплексну модель безпеки для граничних архітектур з використанням агентного підходу. Модель включає в себе агента та периферійний вузол, що функціонує як брокер MQTT та менеджер безпеки. Агент використовує протоколи MQTT та шифрування ChaCha20Poly1305 для безпечного зв'язку, в той час як периферійний вузол займається генерацією токенів, автентифікацією та безперервним моніторингом загроз. NIST Risk Management Framework (RMF) застосовується для систематичного виявлення, оцінки та пом'якшення ризиків безпеки. Ключові ризики включають вразливість апаратного та програмного забезпечення, мережеві загрози та проблеми цілісності даних. Модель реагує на ці ризики за допомогою механізмів безпечного завантаження, апаратних модулів безпеки, надійного шифрування, багатофакторної автентифікації та безперервного моніторингу. Застосовуючи запропоновану модель, можна зменшити загальні ризики, пов’язані з найпоширенішими атаками приблизно в 1.5-2 рази.

Третій розділ присвячений визначенню працездатності моделі, оцінки її роботи, проведенням експериментів у середовищі віртуальної машини з обмеженими ресурсами. Використання процесора та пам'яті записувалося з різною

кількістю клієнтів, а результати були візуалізовані у вигляді графіків, що показують споживання ресурсів. Також окрім продуктивності було проаналізовано міри безпеки моделі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. IEEE Internet of Things Journal, 3(5), 637–646. doi:10.1109/jiot.2016.2579198
2. MEC Support for Edge Native Design https://www.etsi.org/newsroom/press-releases/2250-new-etsi-white-paper-on-mec-support-for-edge-native-design-an-application-developer-perspective
3. P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," in _IEEE Communications Surveys & Tutorials_, vol. 19, no. 3, pp. 1628-1656, thirdquarter 2017, doi: 10.1109/COMST.2017.2682318.
4. Jianli Pan and Zhicheng Yang. 2018. Cybersecurity Challenges and Opportunities in the New "Edge Computing + IoT" World. In Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization (SDN-NFV Sec'18). Association for Computing Machinery, New York, NY, USA, 29–32. https://doi.org/10.1145/3180465.3180470
5. A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider and M. Hamdi, "A Survey on Security and Privacy Issues in Edge-Computing-Assisted Internet of Things," in IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4004-4022, 15 March15, 2021, doi: 10.1109/JIOT.2020.3015432.
6. H. Zeyu, X. Geming, W. Zhaohang and Y. Sen, "Survey on Edge Computing Security," *2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, Fuzhou, China, 2020, pp. 96-105, doi: 10.1109/ICBAIE49996.2020.00027.
7. The Rise of Edge Computing: What it is and Why it Matters. Режим доступу: https://ccsure.com/the-rise-of-edge-computing-what-it-is-and-why-it-matters/
8. Embedded Rust: where are we today? Режим доступу: https://www.embedded.com/embedded-rust-where-are-we-today/

9. NIST Risk Management Framework. Режим доступу: https://csrc.nist.gov/projects/risk-management/about-rmf

10. Мороз, Д. , & Гальчинський, Л. (2024). “Агентна модель для безпечної комунікації в Edge-орієнтованій інфраструктурі”. Collection of Scientific Papers «ΛΌГОΣ», (May 31, 2024; Berlin, Federal Republic of Germany), 170–175. https://doi.org/10.36074/scientia-31.05.2024

11. Quantifying risk https://www.pmi.org/learning/library/quantitative-risk-assessment-methods-9929

12. THE QUANTIFICATION OF RISK http://www.hargreavesrs.co.uk/pdfs/QuantifyingRisk.pdf

13. The generalized risk scale – a scalar integrated tool for developing risk criteria by consensus, in the field of explosives for civil uses Camelia Lavinia Unguras, Doru Anghelache, Victor Gabriel Vasilescu, Florian Stoian and Gabriel Ioan Ilcea MATEC Web Conf., 305 (2020) 00078 DOI: https://doi.org/10.1051/matecconf/202030500078

14. Why Security Teams Need to Focus on Impact Analysis https://www.safeguardcyber.com/blog/security/impact-analysis-cybersecurity-business-imperative

15. Venkatachary, S.K., Alagappan, A. & Andrews, L.J.B. Cybersecurity challenges in energy sector (virtual power plants) - can edge computing principles be applied to enhance security?. Energy Inform 4, 5 (2021). https://doi.org/10.1186/s42162-021-00139-7 https://link.springer.com/article/10.1186/s42162-021-00139-7

16. Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu and W. Lv, "Edge Computing Security: State of the Art and Challenges," in Proceedings of the IEEE, vol. 107, no. 8, pp. 1608-1631, Aug. 2019, doi: 10.1109/JPROC.2019.2918437.

17. T. A. Ahanger, U. Tariq and M. Nusir, "Real-Time Methodology for Improving Cyber Security in Internet of Things Using Edge Computing During Attack

Threats," 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2019, pp. 293-297, doi: 10.1109/ICSSIT46314.2019.8987779.

18. Amandeep Singh Sohal, Rajinder Sandhu, Sandeep K. Sood, Victor Chang, A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments, Computers & Security, Volume 74, 2018, Pages 340-354, ISSN 0167-4048,https://doi.org/10.1016/j.cose.2017.08.016.

19. W. Wu, Q. Zhang and H. J. Wang, "Edge Computing Security Protection from the Perspective of Classified protection of Cybersecurity," 2019 6th International Conference on Information Science and Control Engineering (ICISCE), Shanghai, China, 2019, pp. 278-281, doi: 10.1109/ICISCE48695.2019.00062.

20. Singh, A., Chatterjee, K. & Satapathy, S.C. An edge based hybrid intrusion detection framework for mobile edge computing. Complex Intell. Syst. 8, 3719–3746 (2022). https://doi.org/10.1007/s40747-021-00498-4.

21. What is Industry 4.0? https://www.ibm.com/topics/industry-4-0.

22. Industrial Internet of Things, IIoT https://www.it.ua/knowledge-base/technology-innovation/promyshlennyj-internet-veschej.

23. Купрієнко, А., & Гальчинський, Л. (2023). Агентна модель майнінгу прав доступу в хмарних середовищах. Scientific Collection «InterConf», (184), 482–490. Retrieved from https://archive.interconf.center/index.php/conference-proceeding/article/view/5128.