Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Вознюк Вікторія Іванівна

на здобуття ступеня вищої освіти Бакалавра

Система виявлення вторгнень у мережі CAN-bus в режимі реального час

Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітня програма Кібербезпека

Шифр КРБКБ. 2102142.21.02.05 ПЗ

Виконала студентка 4 курсу група КБ-21-2 Вікторія ВОЗНЮК
Керівник канд. техн. наук, доцент Михайло КАСЯНЧУК
Нормоконтролер старший викладач Сергій МОСТОВИЙ

До захисту допускаю:
Завідувач кафедри кібербезпеки Юрій КЛЬОЦ
11 06 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Рівень вищої освіти Бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ

Завідувач кафедри кібербезпеки

Юрій КЛЬОЦ

15 лютого 2025 р.

З А В Д А Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Вознюк Вікторії Іванівни

1 Тема роботи Система виявлення вторгнень у мережі CAN-bus в режимі реального часу

Керівник роботи Михайло Касянчук

Затверджено наказом ректора університету від 7 лютого 2025 № 23

2 Строк подання студентом кваліфікаційної роботи на кафедру 17.06.2025

3 Вихідні дані до роботи Дослідити особливості функціонування мережі CAN-bus у транспортних засобах. Проаналізувати потенційні вектори атак та загрози безпеці в контексті даного протоколу. Розробити симуляційне середовище у Windows для моделювання CAN-трафіку та атак, реалізувати базову систему виявлення вторгнень у режимі реального часу із використанням Python. Оцінити ефективність побудованої системи.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Аналіз архітектури та особливостей CAN-bus,Класифікація типових атак на CAN (DoS, Spoofing, Replay),Аналіз існуючих методів виявлення атак у CAN,Проєктування тестового симуляційного середовища,Реалізація генерації легітимного та шкідливого трафіку.

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень) «Структура мережі CAN-bus», «Схема поведінкової моделі виявлення атак у мережі CAN-bus», «Алгоритм виявлення вторгнень у CAN-bus мережі»

6 Консультанти розділів кваліфікаційної роботи

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

7 Дата видачі завдання 16 лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Вибір і затвердження теми кваліфікаційної роботи	Лютий	
Аналіз архітектури CAN-bus, типів шин та особливостей функціонування	Лютий	
Дослідження типових атак у мережі CAN (DoS, Spoofing, Interception)	Лютий	
Аналіз існуючих рішень щодо виявлення атак у CAN-bus	Березень	
Формування технічних вимог до майбутньої системи виявлення вторгнень	Березень	
Проєктування архітектури програмної частини системи	Квітень	
Створення симуляційного середовища для генерації CAN-трафіку	Квітень	
Реалізація сценаріїв атак (Replay, Flooding, Spoofing)	Травень	
Реалізація механізмів виявлення аномалій у трафіку	Травень	
Проведення тестування системи, фіксація результатів та аналіз ефективності	Червень	
Оформлення пояснювальної записки та графічного матеріалу, підготовка до захисту	Червень	

Студентка _____ Вікторія ВОЗНЮК

Керівник кваліфікаційної роботи _____ Михайло КАСЯНЧУК

АНОТАЦІЯ

Тема кваліфікаційної роботи: Система виявлення вторгнень у мережі CAN-bus в режимі реального часу.

Автор роботи: Вознюк Вікторія Іванівна.

Керівник роботи: Касянчук Михайло Миколайович.

Пояснювальна записка: 65 с.,3 додатки,9 рисунків,2 таблиці,41 джерел.

Графічна частина: 3 плакати, презентаційних слайдів.

CAN-BUS, ВИЯВЛЕННЯ ВТОРГНЕНЬ, КІБЕРБЕЗПЕКА АВТОМОБІЛІВ, АНАЛІЗ ТРАФІКУ, ДОС АТАКИ, SPOOFING, ВИЯВЛЕННЯ АНОМАЛІЙ, МАШИННЕ НАВЧАННЯ.

Кваліфікаційна робота бакалавра присвячена розробці системи виявлення вторгнень у мережу CAN-bus в режимі реального часу. У роботі проаналізовано основні типи атак на CAN- bus. Визначено ключові вразливості мережі CAN та розглянуто сучасні методи виявлення атак.

У результаті розроблено систему моніторингу CAN-bus, яка аналізує мережевий трафік та ідентифікує потенційні загрози у режимі реального часу. Робота містить алгоритми детектування аномалій, засновані на методах машинного навчання та евристичних підходах. Виконано підготовку до впровадження розробленої системи у транспортні мережі.

9 .06 .2025 _____

ABSTRACT

Subject of qualification work: Real-time intrusion detection system in the CAN-bus network.

Author: Vozniuk Viktoriia Ivanivna.

Head of work: Kasianchuk Mykhailo Mykolaiovych.

Explanatory note:65 p., 3 appendices,9 figures,2 tables, 41 sources

Graphic part: 3 posters, presentation slides.

CAN-BUS, INTRUSION DETECTION, AUTOMOTIVE CYBERSECURITY, TRAFFIC ANALYSIS, DOS ATTACKS, SPOOFING, ANOMALY DETECTION, MACHINE LEARNING.

The bachelor's qualification work is devoted to the development of a real-time intrusion detection system for the CAN-bus network. The study analyzes the main types of attacks on the CAN-bus. The key vulnerabilities of the CAN network have been identified, and modern attack detection methods have been examined.

As a result, a CAN-bus monitoring system has been developed, capable of analyzing network traffic and identifying potential threats in real-time. The work includes anomaly detection algorithms based on machine learning methods and heuristic approaches. Preparations have been made for the implementation of the developed system into vehicle networks.

9.06.2025 _____

ЗМІСТ

ВСТУП

Сучасні автомобільні системи стають дедалі більш інтегрованими та автоматизованими, що зумовлює активне використання внутрішніх мереж обміну даними, серед яких CAN-bus є однією з найпоширеніших. Цей протокол забезпечує швидку та ефективну взаємодію між електронними блоками управління транспортного засобу - такими як блоки управління двигуном, гальмами, освітленням, клімат-контролем, безпековими системами тощо. Однак попри свою ефективність, протокол CAN не має вбудованих засобів захисту інформації, автентифікації чи шифрування, що робить його вразливим до широкого спектру атак.

З кожним роком збільшується кількість кіберінцидентів, спрямованих на перехоплення, модифікацію чи підміну повідомлень у CAN-мережах. Це можуть бути атаки типу DoS, Spoofing, Replay, або більш складні маніпуляції з даними, що загрожують не лише функціональності, а й безпеці водія та пасажирів. Тому розробка ефективних засобів захисту таких мереж — вкрай актуальна задача.

Метою цієї дипломної роботи є створення програмної системи для виявлення атак у CAN-мережі в реальному часі з використанням мови Python. У рамках даного дослідження передбачається моделювання CAN-трафіку, реалізація типових сценаріїв атак, розробка алгоритмів для виявлення аномалій, а також проведення оцінки точності та ефективності розробленої системи. Результати цього дослідження можуть бути використані як фундамент для подальшого вдосконалення захисту автомобільних мереж у галузі транспортної кібербезпеки.

1 АНАЛІЗ НАЯВНИХ РІШЕНЬ

1.1 Мережі CAN-bus

Мережі CAN-bus – це стандарт, призначений для об'єднання різних електронних компонентів в єдину мережу для створення ефективного та надійного обміну даними. Цей стандарт був створений у 1986 році німецькою компанією Bosch. CAN-bus створювався суто для автомобільної промисловості, але з часом почав застосовуватися в багатьох інших галузях [1].

CAN-bus використовується в побутовій техніці, а саме в інтелектуальній системі управління будинком Smart Home. Шина CAN-bus забезпечує обмін сигналами зв’язку між пристроями, дозволяючи їх з’єднувати та здійснювати дистанційне керування. У системі Smart Home до складу компонентів входять панель керування, різноманітні датчики, виконавчі механізми, а також пристрої, що підлягають керуванню. За допомогою панелі керування, яка найчастіше представлена смартфоном, можна не лише управляти електронними пристроями, як-от телевізор чи кавоварка, а й налаштовувати яскравість світла, температуру опалення та контролювати різноманітні системи безпеки. У системі керування розумним будинком, за допомогою CAN-шини, можна повністю контролювати будинок, що значно покращує життя [2].

Також даний стандарт використовується в енергетичній промисловості. Наприклад у сонячних електростанціях, вітрових турбінах. У випадку сонячних панелей, шина CAN виступає сполучною ланкою між основними компонентами системи. Одним з яких є сонячні інвертори, що перетворюють струм з сонячної панелі на змінний струм певної чистоти та напруги, CAN-bus передає стан інформацію цього компонента, а саме про потужність та параметри струму. Ще одним з компонентів з системи є контролери заряду, що заряджає батарею вже переробленим струмом, за допомогою шини можна керувати зарядом акумулятора. У випадку системи моніторингу за допомогою CAN-bus можна зчитати дані про температуру та продуктивність панелей. Також у цій системі є під’єднанні датчики навколишнього середовища, тому можна оптимізувати

роботу панелей залежно від умов освітлення чи температури. У вітрових турбінах шина CAN-bus застосовується для керування різними підсистемами, зокрема системами регулювання лопатей, генераторами та елементами безпеки. Системи контролю лопатей регулюють кут їхнього нахилу залежно від швидкості вітру з метою максимально ефективного вироблення енергії. Інформація про вихідну потужність генератора передається через шину CAN з метою забезпечення стабільного функціонування системи. Сенсори швидкості та напряму вітру, температурні датчики та інші сенсори надають важливі параметри для оптимізації роботи турбіни. Крім того, CAN-bus забезпечує безпеку турбіни, передаючи сигнали про аварійні ситуації, наприклад, при перевищенні швидкості вітру. Центральна система управління використовує отримані дані для віддаленого моніторингу та управління турбіною, а також для діагностики та обслуговування [3-4].

Ще однією сферою застосування шини CAN-bus є медицина, зокрема — медичне обладнання. Прикладом слугує пристрій для моніторингу пацієнта. У даному варіанті CAN-bus використовується для з’єднання різних датчиків які зчитують температуру, пульс, артеріальний тиск, рівень кисню в крові та інші показники, що допомагають лікарям слідкувати за станом пацієнта. За допомогою CAN-bus всі ці дані передаються до центрального комп’ютера, який обробляє та аналізує всі ці дані [5].

Основна галузь де використовується CAN-bus є автомобільна промисловість. За допомогою цієї шини з’єднані різні електроні компоненти та системи в автомобілі, що забезпечує надійну та ефективну роботу. CAN-bus дає можливість передавати дані між різними модулями та сенсорами в реальному часі, забезпечуючи оптимальну роботу складних систем автомобіля таких складних систем, як двигун, трансмісія, гальмівна система, рульове управління і навіть клімат-контроль. Якщо порівнювати це з людським організмом, то CAN-bus можна уявити як нервову систему, що передає інформацію між частинами тіла, а електронні блоки управління - це як частини тіла, з’єднані між собою. Інформацію, що сприймається однією частиною, можна передавати іншим

частинам для належної реакції, що дозволяє ефективно координувати роботу різних систем автомобіля. ECU, вони ж «CAN-вузли», підключаються через двопрохідну шину, що складається з дротів CAN high і CAN low, які передають диференційовані сигнали. Це дозволяє знизити кількість проводів в автомобілі, що робить конструкцію більш компактною і зручною. Крім того, таке з’єднання гарантує надійний обмін даними навіть за наявності електромагнітних завад, що є критично важливим для стабільного функціонування всіх систем автомобіля [6].

Мережа CAN-bus має спрощену структуру, що дозволяє ефективно обмінюватися даними між різними компонентами. Вона складається з декількох електронних блоків управління, які можуть бути позначені як "ECU 1", "ECU 2" та "ECU 3". Ці блоки з'єднані між собою двома паралельними лініями - проводом CAN High та проводом CAN Low. Стрілки на діаграмі вказують на двонаправлений потік даних, підкреслюючи взаємний обмін інформацією між усіма ECU, що дозволяє мережевим компонентам «спілкуватися» та координувати свою роботу без участі центрального комп'ютера.

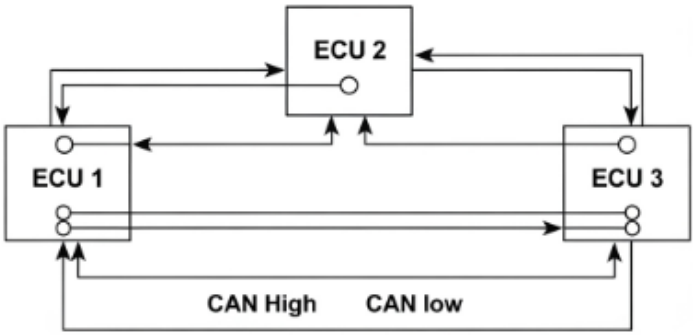


Рисунок 1.1 - Структура мережі CAN-bus

Мережа CAN працює без потреби в центральному комп'ютері та дозволяє кільком електронним блокам керування ECU обмінюватися повідомленнями, забезпечуючи швидку та ефективну роботу. Наприклад, шина CAN забезпечує обмін інформацією між двигуном та гальмами, що дозволяє адаптувати роботу систем у реальному часі залежно від умов дорожнього руху та стану автомобіля.

У сучасних автомобілях може бути понад 70 ECU, і кожен з них обмінюється даними з іншими через CAN- bus. Це сприяє оптимізації функціонування всіх ключових систем автомобіля.

Всі ECU складаються з трьох основних елементів: мікроконтролера, CAN-контролера та CAN-трансфер. Мікроконтролер інтерпретує вхідні CAN-повідомлення і визначає, які дані передавати далі. CAN-контролер, як правило, інтегрований в мікроконтролер і забезпечує правильну комунікацію по шинах CAN, виконуючи кодування повідомлень, виявлення помилок та арбітраж передачі. Канальний трансфер з'єднує контролер з фізичними проводами шини, перетворюючи дані в диференційні сигнали для шини та забезпечуючи необхідний захист від електричних перешкод.

Таким чином, система CAN в автомобілі є критично важливою для забезпечення належної роботи та взаємодії всіх компонентів. Вона дозволяє швидко передавати важливі дані між системами, забезпечуючи ефективну роботу всіх функцій автомобіля [7].

Існує кілька різновидів шини CAN-bus, які застосовуються залежно від конкретних вимог і відрізняються між собою швидкістю передачі даних, рівнем надійності та іншими технічними характеристиками. Високошвидкісні мережі CAN є більш поширеними, оскільки вони пропонують швидкість передачі до 1 Мбіт/с, що достатньо швидко для підтримки складних та високошвидкісних систем транспортних засобів, таких як двигун, трансмісія, гальма та рульове керування. Це дає змогу оперативно передавати критично важливі дані в режимі реального часу, забезпечуючи своєчасну реакцію на зміни в роботі автомобіля. Вони здатні функціонувати навіть в умовах значних електричних завад, що характерні для автомобільного середовища. Це забезпечує стабільність та надійність в роботі, що критично важливо для безпеки автомобіля. Високошвидкісні CAN-мережі забезпечують швидкий обмін даними, що робить їх широко застосовуваними в багатьох сучасних автомобільних системах. Це сприяє зменшенню кількості проводів і спрощенню інтерфейсів, що робить автомобіль більш інтегрованим і зручним в експлуатації.

Низькошвидкісний CAN застосовується в автомобілях та інших системах, де пріоритетом є надійність і стійкість до збоїв, тоді як висока швидкість передачі даних не є критичною. Вона працює з максимальною швидкістю до 125 кбіт/с, що достатньо для багатьох систем, де швидка передача даних не є критично важливою. Це включає такі компоненти, як сидіння, дзеркала, освітлення, дверні замки тощо. Низькошвидкісна CAN мережа працює навіть у разі часткових пошкоджень або несправностей, може безперебійно передавати дані навіть в умовах збоїв в окремих елементах мережі, що підвищує загальну надійність автомобіля. Подібно до високошвидкісних CAN-систем, низькошвидкісна шина також використовує двопровідне підключення, що допомагає скоротити кількість необхідних проводів у автомобільній мережі. Це також сприяє зниженню вартості та спрощенню процесу встановлення.

CAN FD - це вдосконалена версія стандарту CAN, яка забезпечує кращу швидкість передачі даних, до 8 Мбіт/с, що перевищує максимальну швидкість класичного CAN 1 Мбіт/с, та можливість передавати більші обсяги інформації, до 64 байт даних в одному повідомленні. Особливістю є можливість застосування різних швидкостей передачі даних на різних етапах роботи системи. При визначенні пріоритету доступу пристроїв до шини використовуються повільніші швидкості до 1 Мбіт/с, тоді як під час безпосередньої передачі даних швидкість збільшується до 8 Мбіт/с. Це дозволяє швидко передавати великі обсяги інформації. CAN FD забезпечує зворотну сумісність з класичними системами CAN, що дозволяє використовувати нові пристрої в існуючих мережах без значних змін.

CAN XL є новіший стандарт, який є наступним етапом розвитку CAN FD. Він забезпечує передачу більших обсягів даних, ніж CAN FD, що робить його ідеальним для систем із підвищеними вимогами до пропускної здатності, а також підтримує ще вищі швидкості передачі. Даних, наближаючись до швидкостей, характерних для сучасних мереж Ethernet, наприклад, 100BASE-T1, що також дозволяє покращити продуктивність при високих вимогах до пропускної здатності. CAN XL як і CAN FD зберігає сумісність із попередніми версіями

CAN, що також дозволяє використовувати нові пристрої в існуючих мережах без значних змін та допомагає поступово інтегрувати нові системи в чинні інфраструктури, забезпечуючи плавний перехід до більш потужних технологій.[8]

CAN-bus має низку важливих аспектів, які визначають її ефективність у різних застосуваннях. Мережа побудована за схемою шинної топології, що передбачає підключення всіх пристроїв до однієї спільної лінії зв'язку.Це дозволяє зменшити кількість проводів і спростити монтаж, що є важливим, наприклад, у транспортних засобах. У CAN-bus для передачі даних використовуються всього дві пари проводів, що сприяє зниженню вартості та ваги системи порівняно з іншими методами, які потребують більшої кількості кабелів. Мережа працює за принципом пріоритетів повідомлень. Кожне повідомлення має ідентифікатор, який визначає його важливість. Якщо кілька пристроїв одночасно намагаються передати дані, пріоритет отримує повідомлення з найнижчим ідентифікатором, що допомагає уникнути конфліктів на шині. CAN-bus також має вбудовані механізми для виявлення помилок. Якщо передане повідомлення пошкоджене, система автоматично ініціює повторну передачу для забезпечення коректної роботи. Мережа має можливість налаштовувати швидкість передачі даних, що дозволяє адаптувати її до різних вимог системи. У випадку необхідності, CAN-bus забезпечує механізм арбітражу, за допомогою якого визначається, яке повідомлення буде передано першим, якщо кілька пристроїв намагаються передати дані одночасно. Крім того, CAN-bus має обмеження по довжині кабелю та кількості пристроїв, які можуть бути підключені до мережі, хоча і для більшості застосувань цього достатньо. Мережа CAN-bus підтримує також різні стандарти протоколів, які допомагають інтегрувати її в більші системи та мережі [9].

Важливою частиною системи CAN є структура повідомлень, яка відповідає за ефективний зв'язок між пристроями. Вона складається з кількох полів, таких як ідентифікатор, контрольне поле, поле даних і механізм перевірки помилок. Ідентифікатор визначає пріоритет повідомлення в мережі. У

стандартних 11-бітових ідентифікаторах CAN 2.0A можливі 2048 різних пріоритетів, а у розширених 29-бітових CAN 2.0B є ще більше можливостей із пів мільярда варіантів. Код довжини даних DLC, вказує на кількість байтів, які містить поле даних, і може бути в межах від 0 до 8 байтів. Поле даних включає фактичну інформацію, яка передається між вузлами. Для перевірки на помилки використовується механізм циклічної ремонтантності CRC, що забезпечує надійність передачі та ініціює повторну передачу в разі виявлення помилки. Слот підтвердження, який складається з одного біта, використовується приймачами для підтвердження успішного прийому повідомлення або сигналізації про помилку. Також передбачене поле помилкового кадру, яке дає можливість вузлам повідомляти про проблеми з передачею або прийомом повідомлень [10].

CAN-bus має свої переваги та недоліки. Деякими з головних плюсів є її здатність забезпечувати швидкий й надійний обмін даними між пристроями, що дуже важливо, наприклад, для автомобільних систем або промислових машин. Однією з основних переваг CAN є економічність, вона використовує лише два дроти для передачі даних, що значно знижує витрати і зменшує вагу обладнання. Це особливо актуально для автомобілів, де кожен зайвий кілограм ваги може впливати на їхню ефективність. Ще однією важливою перевагою є надійність мережі. CAN оснащений вбудованими механізмами для виявлення помилок. Наприклад, якщо передача даних не була успішною через помилку, система автоматично намагається надіслати повідомлення знову. Якщо якийсь пристрій мережі постійно передає помилки, його можна відключити, що допомагає запобігти виникненню проблем у всій мережі. Така система дає можливість швидко виявляти та виправляти несправності, що робить мережу надійною. CAN-bus вирізняється гнучкістю, оскільки кожен пристрій у мережі має власний контролер, що забезпечує легке та швидке додавання або заміну пристроїв без суттєвих змін у загальній архітектурі системи. Мережа CAN надає централізовану точку керування для всіх пристроїв, що полегшує діагностику та налаштування системи. Крім того, за допомогою пріоритетності повідомлень система здатна надавати доступ до шини для найбільш важливих повідомлень

без затримок, що є великим плюсом у реальному часі.

Проте, як і будь-яка технологія, CAN має і свої недоліки. Одним з основних обмежень є максимальна кількість пристроїв, які можуть бути підключені до мережі у такому випадку лише 64 вузли. Для деяких великих систем цього може бути недостатньо. Крім того, довжина кабелю CAN-bus обмежена 40 метрами, що також створює труднощі для дуже великих мереж, наприклад, в транспортних засобах чи великих промислових об'єктах. Це обмеження можна подолати за допомогою спеціальних підсилювачів сигналу, але це ускладнює систему. Також стандартна швидкість передачі даних в мережі CAN складає 1 Мбіт/с, для складніших задач цього може бути недостатньо, але це обмеження було вирішено в розширеній версії CAN FD, яка забезпечує швидкість до 8 Мбіт/с. Ще один недолік CAN-bus - це електричні шуми, які можуть виникати через різницю напруги між пристроями. Ці шуми можуть заважати іншим системам, що працюють в тій самій мережі, і це потрібно враховувати при проєктуванні. Створення програмного забезпечення для налаштування та управління мережею CAN вимагає значних фінансових вкладень, що може бути перешкодою для невеликих компаній або нових проєктів.

Попри ці недоліки, CAN залишається одним з найпопулярніших та найефективніших варіантів для створення надійних мереж, особливо в таких сферах, як автомобільна промисловість, промислова автоматизація та енергетика. Це дуже економічне і зручне рішення, яке забезпечує стабільну і швидку передачу даних, але потребує уважного підходу до планування мережі та врахування її обмежень [11].

1.2 Поширені типи атак у CAN

CAN широко використовується в автомобільних мережах, адже вона ідеально підходить для швидкого обміну даних між різними сенсорами та виконавчими механізмами. Тобто завдяки мережі CAN всі компоненти мережі

отримують повідомлення вчасно або не отримують взагалі. Однак мережа CAN має і свої мінуси, основним з яких є те що вона не була розроблена з урахуванням вимог безпеки, оскільки CAN не має вбудованих механізмів аутентифікації та шифрування, він вразливий до різних типів атак. Різні дедалі системи все частіше йдуть з підключенням до інтернету, які і є об’єктом для атак. Наприклад, зловмисник може примусово активувати подушки безпеки, надіславши спеціальні діагностичні команди через CAN. Саме тому захист цієї шини є надзвичайно важливим.

Основні типи атак на CAN-bus можна поділити на три категорії: атаки відмови в обслуговуванні - DoS, атаки підміни - Spoofing, атаки на перехоплення та маніпуляцію кадрами - Interception & Manipulation Attacks.

Атаки відмови в обслуговуванні – це атаки при яких зловмисник робить систему або дані недоступними для їхніх справжніх користувачів. Хакер намагається заблокувати доступ до мережі, системи або комп'ютера, перевантажуючи їх фальшивими запитами або надмірним трафіком. Це перешкоджає нормальній роботі користувачів, викликаючи уповільнення роботи або навіть повне відключення системи [12].

Одним з прикладом DoS атак на мережу CAN-bus є атака на захоплення шини - Bus Flood Attack. Під час даної атаки зловмисник блокує роботу системи, передаючи кадри CAN дуже швидко, забираючи всю пропускну здатність шини. Через це легітимні кадри можуть затримуватися, а деякі частини системи можуть не отримувати кадри вчасно, що призводить до збоїв. Успішність атаки залежить від наявних механізмів захисту. Якщо шина не має обмежень, кадр з CAN ID 0 заблокує весь інший трафік, оскільки цей індикатор має найвищий пріоритет. Якщо ж є спеціальний шлюз, який пропускає тільки певні CAN ID, тоді атака торкнеться лише кадрів з нижчим пріоритетом, а кадри з вищим пріоритетом продовжать передаватися без перешкод. Саме тому стандартні діагностичні кадри ODB-II мають CAN ID 0x7df і вище - це дає їм низький пріоритет.

Наступним прикладом атаки відмови в обслуговуванні на мережу CAN-bus є атака "Bus-off". Ця атака спрямована на виведення з ладу конкретного ECU -

електронного блоку управління, не порушуючи роботу інших пристроїв у мережі. Під час атаки зловмисник змушує націлений ECU перейти в режим "bus-off", що означає відключення цього блоку від шини. Механізм атаки полягає в тому, що зловмисник генерує помилки в передачі даних, що змушує ECU накопичувати помилки в лічильнику помилок. Коли лічильник досягає певного порогу, наприклад, 255, ECU автоматично відключається від шини. Це позбавляє систему можливості обробляти або передавати дані через цей блок. Важливою рисою атаки є те, що деякі ECU можуть спробувати відновитися автоматично. Проте, якщо атака повторюється, а система має відповідні механізми управління мережею та діагностики, вона може заблокувати спроби відновлення і встановити спеціальний прапорець в пам'яті пристрою, що змусить його залишатися офлайн. Це може призвести до значного зниження функціональності, наприклад, активації обмеженого режиму роботи автомобіля, такого як "Limp Home", який дозволяє обмежено керувати транспортним засобом, але знижує його потужність для запобігання подальшим ушкодженням. У випадку, якщо мета атаки — створити проблеми для користувачів або порушити роботу транспортних засобів, така атака вважається успішною. Вона може бути використана не тільки для виведення окремого ECU з ладу, а і як частина складніших атак, спрямованих на саботаж або маніпуляцію роботою транспортних систем [13].

Одним із прикладів атаки відмови в обслуговуванні на мережу CAN-bus є атака «Freeze Doom Loop». Робота даної атаки полягає в тому щоб заморозити трафік шини на деякий час, що може призвести до затримки кадрів або зменшення пропускної здатності шини в цілому. Протокол CAN визначає домінантний біт у першому біті проміжку між кадрами IFS, який сигналізує про перевантаження контролера. У старих версіях протоколу це дозволяло повільним контролерам отримати більше часу для обробки кадрів, не змінюючи лічильників помилок. Зловмисник використовує цю особливість генеруючи домінантний біт на піні CAN TX в першому біті IFS, після чого спостерігає за процесом відновлення помилок. Потім зловмисник повторно генерує домінантний біт в

полі IFS після завершення цього процесу, що призводить до ще більшої затримки. Атака може повторюватися стільки разів, скільки потрібно, ефективно заморожуючи шину. Це може призвести до затримок передачі кадрів, що негативно впливає на роботу системи, але не збільшує лічильників помилок, що ускладнює виявлення атаки. Основна складність цієї атаки в тому, що вона дуже важко діагностується. Єдина ознака - це затримка кадрів, що може бути прийнято як звичайний тимчасовий збій. Оскільки лічильники помилок не змінюються, а кадри приходять із затримкою, атака може залишитися непоміченою, якщо система не проводить детальний аналіз латентності передачі кадрів [14].

Наступний тип атак на CAN-bus є атаки на підробку кадрів - Spoofing. Під час даних атак зловмисник створює фальшиві кадри, щоб обманути систему, змусивши її прийняти ці кадри за легітимні. Це може призвести до різних збоїв у роботі мережі CAN, таких як маніпулювання даними або виведення з ладу певних систем [15].

Simple Frame Spoofing є одним з видів спуфінгових атак. Під час атаки зловмисник створює і відправляє кадри, що виглядають як звичайні кадри мережі CAN. Метою цієї атаки є обман системи, видаючи фальшиві дані за справжні. Атака може бути націлена на конкретні пристрої в мережі, щоб змусити їх виконувати небажані або шкідливі операції. Процес цієї атаки простий, зловмисник створює фальшивий кадр з конкретним CAN ID, який виглядає як справжній кадр від іншого ECU в мережі. Оскільки CAN протокол не має механізмів автентифікації кадрів, приймач не може відрізнити підроблений кадр від легітимного. Це дозволяє атакуючому змінювати або маніпулювати даними в мережі, що може мати серйозні наслідки, такі як некоректна робота системи або навіть саботаж. Для успішного виконання цієї атаки зловмисник повинен мати доступ до шини CAN та знати структуру кадрів, які використовуються в системі. Оскільки CAN не містить вбудованих механізмів захисту від підробки, підроблені кадри можуть бути прийняті системою як справжні, що робить цю атаку досить небезпечною [16].

Також однією зі спуфінгових атак є адаптивне підроблення - Adaptive

Spoofing. Під час даної атаки зловмисник намагається підробити кадри таким чином, щоб вони не викликали конфліктів з легітимними кадрами, що передаються в мережі. У цій атаці атакуючий пристрій слухає шину, щоб дізнатися, коли з'являється легітимний кадр, і тоді підготовляє підроблений кадр. Замість того, щоб одразу передавати підроблений кадр, зловмисник чекає, коли легітимний кадр вже буде в мережі, і відправляє свій підроблений кадр так, щоб він не перешкоджав передаванню справжнього. Це важливо, оскільки якщо підроблений кадр буде надісланий відразу після легітимного, то він може перезаписати дані в буфері приймача, і той обробить підроблений кадр замість справжнього. Цей тип атаки складніший, тому що зловмисник повинен точно синхронізувати свої дії з таймінгом мережі. Приймач має дуже коротке вікно часу для того, щоб зберегти кадр в буфері перед його обробкою. Зловмисник повинен надіслати підроблений кадр у це вузьке вікно часу, щоб "заманити" систему і змусити її прийняти підроблений кадр замість справжнього. Це дає атакуючому змогу маніпулювати даними в мережі, не порушуючи порядок передачі кадрів, що робить таку атаку важчою для виявлення [17].

Наступною атакою є Error Passive Spoofing Attack. Атака використовує специфічну особливість протоколу CAN, щоб обійти механізми виявлення помилок і підробити кадри. Ця атака є складнішою і важко виявляється, оскільки вона спирається на вразливість в обробці помилок самим контролером CAN. Роботу даної атаки можна поділити на два етапи.

Першим етапом буде переведення пристрою в режим "Error Passive". Спочатку зловмисник генерує помилки на шині, щоб змусити ECU - електронний блок управління перейти в Error Passive режим. В цьому режимі контролер не може правильно сигналізувати про помилки, оскільки він не може сам ініціювати повідомлення про помилки. Замість цього, він чекає, поки інші пристрої в мережі зафіксують помилку і сигналізують про неї. Кожного разу, коли на шині виникає помилка, лічильник помилок цього ECU збільшується, поки не досягне критичного рівня, після чого пристрій переходить в Error Passive режим.

Другим етапом буде підслуховування та підробка кадрів. Після того як

контролер потрапляє в цей режим, зловмисник починає підслуховувати шину, щоб дізнатися, який кадр виграє доступ до шини й почне передаватися. Коли атакуючий помічає, що легітимний кадр був прийнятий і отримає доступ до шини, він миттєво генерує підроблений кадр з тим самим CAN ID і заміщує справжній кадр. Завдяки тому, що контролер знаходиться в Error Passive режимі, він не може сам сигналізувати про помилку, і підроблений кадр проходить як легітимний.

Зловмисник змінює не тільки дані кадру, але й CRC - перевірку на помилки, щоб його кадр виглядав абсолютно правильно, і приймач не зміг помітити підробку. Таким чином, система прийме фальшиві дані, що може призвести до серйозних збоїв у роботі пристроїв.

Дану атаку дуже важко виявити, адже вона не викликає очевидних помилок у мережі. Коли зловмисник підміняє кадри, вони виглядають як звичайні, правильні кадри для шини. Контролери не можуть сигналізувати про помилки, тому система не помічає, що щось не так. Зловмисник ретельно слідкує за мережею, чекаючи моменту, коли буде надіслано легітимний кадр, і в цей момент підміняє його на фальшивий. Таке підмінювання не порушує роботу інших пристроїв і не викликає збоїв у системі. Для того, щоб помітити таку атаку, потрібно використовувати спеціальні інструменти, які здатні дуже детально перевірити кадри в мережі. Оскільки звичайні методи моніторингу не помічають таких змін, ця атака є дуже небезпечною і важкою для виявлення [18].

Атаки на перехоплення та маніпуляцію кадрами, Interception & Manipulation Attacks - це атаки, коли зловмисник перехоплює або змінює вже надіслані кадри в мережі CAN. Він може повторно надіслати старий кадр або змінити частини кадру, щоб маніпулювати інформацією, яку отримують інші пристрої. Це може спричинити помилки або небажану поведінку системи, наприклад, змусити ECU працювати з неправильними даними або виконувати небажані команди. Такі атаки можуть бути складними для виявлення, оскільки вони не порушують загальний потік кадрів, але змінюють їх зміст [19].

Double Receive Attack це атака на перехоплення та маніпуляцію кадрами.

Дана атака використовує незначну уразливість у процесі прийманні кадрів на CAN-bus. Коли передавач надсилає кадр, він визначає його завершення по останньому біті, а приймач - по передостанньому. Якщо в останньому біті трапляється помилка, передавач повторно надсилає той самий кадр. Проблема в тому, що приймач уже прийняв цей кадр і передав його системі. Тому, коли кадр надсилається знову, приймач його приймає вдруге, хоча система вже обробила перший кадр. Зловмисник може використати це, щоб змусити систему прийняти один і той самий кадр два рази. Це дозволяє йому маніпулювати даними або дублювати важливі команди, що може викликати збої або неправильну роботу пристроїв у мережі CAN [20].

Атаки на мережу CAN можуть мати серйозні наслідки для автомобільних систем, зокрема, роблячи їх вразливими до маніпуляцій. Це може загрожувати безпеці водія, пасажирів і пішоходів. Атаки відмови в обслуговуванні (DoS) можуть серйозно порушити роботу автомобіля. Наприклад, атака на захоплення шини може блокувати передачу важливих даних між електронними блоками управління, що може призвести до збоїв в системах, таких як система антиблокувального гальмування або подушки безпеки, що створює небезпеку на дорозі. Атаки на підробку кадрів дозволяють зловмисникам видавати фальшиві дані за справжні. Це може призвести до помилкових команд для ECU, таких як вимкнення двигуна або зміна налаштувань автомобіля, що може бути небезпечним для водія і дорого коштувати на ремонті. Атаки на перехоплення та маніпуляцію кадрами можуть змусити систему прийняти неправильні дані, що призведе до несанкціонованого включення або вимикання важливих систем, таких як стабілізація або гальмування, що може викликати аварії.

Найскладнішою і небезпечнішою атакою є Error Passive Spoofing Attack**,** адже вона дозволяє зловмиснику замінювати кадри без того, щоб система помітила підміну. Оскільки підроблені кадри виглядають як справжні, система не розпізнає атаку і продовжує працювати з фальшивими даними. Це робить атаку надзвичайно небезпечною і складною для виявлення, що може призвести до серйозних збоїв або навіть аварії.

1.3 Постановка задачі

CAN-bus є важливою складовою сучасних транспортних систем, що забезпечує взаємодію між електронними блоками управління у режимі реального часу. Незважаючи на свою популярність, високу швидкодію та надійність, протокол CAN не передбачає вбудованих механізмів автентифікації, шифрування або контролю доступу. Внаслідок цього він залишається вразливим до цілого спектру атак, зокрема атак типу DoS, Spoofing, Replay, Interception та Manipulation. Це створює серйозну загрозу безпеці як для самого транспортного засобу, так і для пасажирів.

З огляду на зростання кількості інцидентів, пов’язаних з хакерськими втручаннями у CAN-мережі, особливо актуальною є проблема розробки ефективної системи виявлення вторгнень, здатної виявляти аномальні дії в мережі в режимі реального часу. Більшість існуючих рішень або орієнтовані на апаратну реалізацію, що ускладнює розгортання у віртуальному середовищі, або ж мають обмежену здатність до адаптації під нові види атак.

Метою роботи є розробка та тестування програмної системи виявлення атак у мережі CAN-bus у режимі реального часу на базі Python, яка функціонує у віртуальному середовищі, без потреби у фізичних CAN-пристроях.

Для реалізації мети було окреслено наступні основні задачі:

– проаналізувати архітектуру та особливості функціонування мережі CAN-bus, зокрема визначити типи CAN-bus, таких як High-speed, Low-speed, та CAN FD, описати структуру кадру та логіку арбітражу, а також виділити переваги і недоліки протоколу з точки зору інформаційної безпеки;

– провести класифікацію та аналіз поширених типів атак у мережі CAN, зокрема DoS-атак, а саме Flooding, Bus-off, Freeze Doom Loop, Spoofing-атак, наприклад Simple, Adaptive, Error Passive, а також атак перехоплення і маніпуляції - Double Receive, Interception;

– проаналізувати існуючі методи виявлення атак у CAN-мережі, включаючи IDS, що базуються на сигнатурах, IDS на основі аномалій, гібридні

системи, а також програмні та апаратні реалізації;

– розробити тестове середовище у операційній системі Windows на базі Python, яке дозволяє імітувати обмін CAN-пакетами без використання фізичних пристроїв, а також генерувати як легітимний трафік, так і трафік із шкідливими впливами, таких як Replay, Flooding, Spoofing;

– реалізувати базові сценарії атак, зокрема Replay-атаку, тобто відтворення раніше записаних повідомлень, Flooding-атаку - генерація великої кількості повідомлень, Spoofing - імітація повідомлень від неавтентичного ECU;

– розробити базову логіку виявлення аномалій у CAN-трафіку, що включає виявлення повторюваних кадрів – Replay-атаки, надмірної частоти повідомлень від одного джерела - Flooding, а також появу нових невідомих ідентифікаторів, тобто Spoofing;

– оцінити результати тестування розробленої системи, зокрема провести аналіз виявлених атак, визначити кількість хибних спрацювань, а також окреслити обмеження симуляційного підходу.

Таким чином, практичним результатом кваліфікаційної роботи має стати повноцінне симуляційне середовище для тестування CAN-мережі, реалізоване виключно у програмному середовищі без потреби у фізичному обладнанні. Основним компонентом проєкту виступає програмна система виявлення атак у CAN-мережі, яка здатна в режимі реального часу аналізувати потік CAN-повідомлень, ідентифікувати характерні ознаки вторгнення та сигналізувати про загрозу. Система передбачає базову логіку виявлення типових атак, таких як Replay, Flooding та Spoofing, і може бути легко адаптована для подальшого розширення чи інтеграції з іншими інструментами кіберзахисту. Крім того, до складу результатів входить повна документація, що описує процес моделювання атак, налаштування симуляційного середовища, реалізовану логіку виявлення аномалій, а також результати тестування із зазначенням рівня точності, кількості хибнопозитивних спрацювань і виявлених обмежень. Це забезпечує можливість відтворення експериментів, їх подальшого вдосконалення та використання в навчальних, дослідницьких або прикладних цілях.

2 МОДЕЛЬ ТА МЕТОД ВИЯВЛЕННЯ ВТОРГНЕНЬ У МЕРЕЖІ CAN-BUS

2.1 Модель виявлення вторгнень у мережі CAN-bus

Проєктування ефективної системи виявлення вторгнень IDS у мережі CAN-bus передбачає розробку математичної та логічної моделі, здатної оперативно ідентифікувати потенційні атаки в режимі реального часу. Протокол CAN, попри свою поширеність у автомобільних і промислових системах, має суттєві обмеження в контексті інформаційної безпеки. Зокрема, він не передбачає механізмів автентифікації джерела повідомлення, відсутнє шифрування даних, а також немає жодної перевірки цілісності і походження кадру, окрім базової контрольної суми. Через це зловмисники можуть безперешкодно проводити різні типи атак, такі як Flooding, Replay або Spoofing, не порушуючи при цьому формальну структуру пакету CAN.

Традиційні методи перевірки цілісності кадрів, наприклад, контрольна сума або перевірка довжини даних, у цій ситуації виявляються неефективними, оскільки вони спрямовані лише на виявлення випадкових помилок, а не навмисної шкідливої активності. Тому для забезпечення надійного захисту необхідно застосувати поведінкову модель, яка зможе аналізувати аномалії в мережевому трафіку, виявляючи відхилення від нормальної роботи системи.

Запропонована модель виявлення атак базується на принципах аналізу аномалій. Цей підхід передбачає спочатку побудову профілю нормального стану системи – типових характеристик трафіку, а вже потім, під час роботи системи, фіксацію значущих відхилень від цього профілю. У випадку CAN-bus це означає визначення типових параметрів, таких як частота передачі повідомлень, характерна послідовність ідентифікаторів (ID) кадрів, середні інтервали часу між їх надходженням, а також щільність трафіку у визначеному часовому вікні. На основі цих характеристик формується набір ознак, які система постійно моніторить [22-23].

Основними компонентами моделі є:

– буферизація кадрів у часовому вікні;

– обчислення метрик;

– побудова порогових значень.

Всі CAN-повідомлення, що надходять у межах заданого інтервалу часу (наприклад, 1000 мс), збираються та зберігаються для подальшого аналізу. Це дозволяє оцінити поведінкові патерни трафіку за короткий період.

Для накопичених кадрів визначаються такі параметри, як середня кількість повідомлень на секунду, кількість унікальних ID, частота появи кожного конкретного ID, а також стандартне відхилення інтервалів між послідовними повідомленнями з одного і того ж ID. Ці показники дають змогу розпізнати аномальні зміни у поведінці мережі.

Визначення меж, за якими вважається, що поведінка системи відхилилася від норми. Порогові значення можуть задаватися вручну на основі попередніх досліджень або формуватися автоматично у фазі навчання системи, коли вона адаптується під конкретне середовище експлуатації

На основі описаних метрик модель здатна виявляти типові ознаки відомих атак:

– flooding-атака;

– replay-атака;

– spoofing-атака.

Характеризується різким і значним збільшенням кількості кадрів з однаковим ID. Наприклад, якщо за нормальних умов повідомлення з певним ID передаються десятки разів на секунду, то при атаці Flooding ця кількість може збільшитись у кілька разів, що призводить до перевантаження шини і відмови легітимних вузлів у передачі даних.

Відзначається надходженням повторюваних кадрів з однаковим ID і тим самим вмістом даних у фіксовані інтервали часу. При цьому відтворюваний трафік не залежить від реального стану системи, що робить його аномальним.

Виявляється через появу в мережі кадрів з ID, які не використовуються в

нормальних умовах, або через зміни у характері даних, що передаються конкретними вузлами.

Особливістю запропонованої моделі є те, що вона не потребує аналізу вмісту поля даних кожного кадру, а фокусується виключно на поведінкових параметрах трафіку, що забезпечує універсальність і здатність виявляти різні типи атак, навіть без їхнього попереднього вивчення. Це особливо важливо в контексті CAN-bus, де ресурсні обмеження і вимоги до низької затримки передачі даних не дозволяють застосовувати складні криптографічні методи або обробку великих обсягів даних. На рисунку 2.1 зображено загальну структуру поведінкової моделі виявлення атак у мережі CAN-bus.

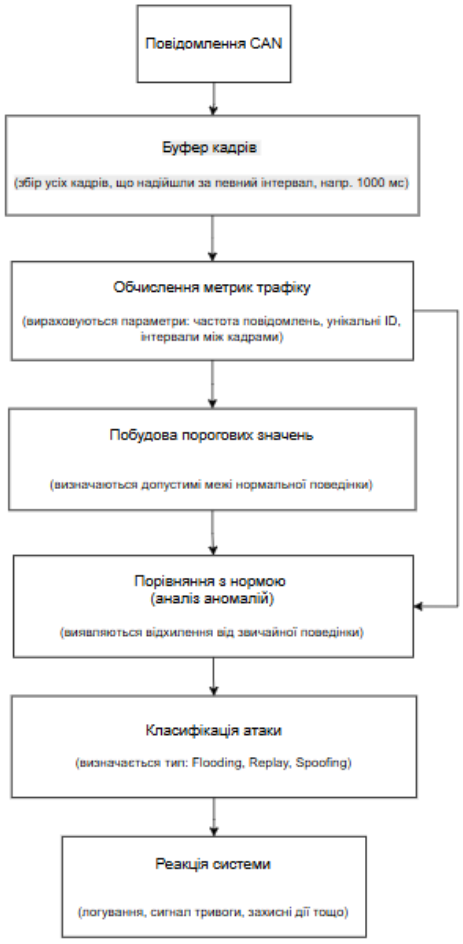


Рисунок 2.1 – Схема поведінкової моделі виявлення атак у мережі CAN-bus

Завдяки обмеженому набору обчислень та компактному представленню метрик, модель може працювати у режимі реального часу, що є критично

важливим для забезпечення безпеки автомобільних систем і промислових мереж на базі CAN.

Отже, описана поведінкова модель створює надійну і практичну основу для побудови систем виявлення атак у мережах CAN-bus, які відповідають ключовим вимогам до таких систем: мінімальна затримка обробки, низьке навантаження на ресурси та висока ефективність виявлення широкого спектру атак.

2.2 Аналіз наявних рішень

Захист мережі CAN-bus від кібератак є важливим завданням, оскільки ця шина зв'язку широко використовується в автомобільній промисловості, промисловій автоматизації, медичних пристроях та інших критичних системах. У зв'язку з відкритістю протоколу CAN та відсутністю вбудованих механізмів аутентифікації та шифрування, мережі CAN-bus є вразливими до різноманітних кібератак, таких як спуфінг, відмова в обслуговуванні та інші. Тому розробка ефективних систем виявлення вторгнень є необхідною для забезпечення безпеки та надійності цих мереж.

На сьогодні існують декілька основних підходів до виявлення атак у мережах CAN-bus, кожен з яких має свої переваги та недоліки.

Один із таких підходів - це методи на основі сигнатур - Signature-based IDS). Цей метод працює за принципом порівняння мережевого трафіку з уже відомими шаблонами атак. Ідея полягає в тому, що кожна відома атака має певні характерні ознаки або шаблони, які можна зафіксувати та використовувати для її виявлення в майбутньому.

У випадку з мережами CAN-bus, де кількість стандартних атак обмежена, такі системи можуть швидко та точно виявляти загрози, які вже відомі. Мережі CAN-bus часто використовуються в критичних системах, таких як автомобільні мережі, де важливо вчасно виявити та зреагувати на можливі загрози, адже будь-

яка затримка може мати серйозні наслідки для безпеки.

Принцип роботи системи на основі сигнатур дуже простий. Система постійно аналізує трафік мережі, збираючи інформацію про передані пакети даних. Кожен такий пакет порівнюється з уже зафіксованими шаблонами атак у базі даних. Якщо пакет даних збігається з шаблоном з бази, система сигналізує про можливу атаку. Це дозволяє швидко виявляти відомі загрози та зупиняти їх [24].

Однією з головних переваг методів на основі сигнатур є те, що вони мають низьку ймовірність хибних спрацьовувань. Це означає, що система спрацьовує тільки в тому випадку, коли є точний збіг з відомим шаблоном атаки. У мережах CAN-bus, де трафік обмежений і структурований, це важливо, оскільки зменшується кількість помилкових спрацьовувань, що дозволяє знижувати навантаження на систему та оператора.

Ще одна важлива перевага сигнатурних систем - це їхня швидкість. Процес порівняння пакунків з шаблонами є простим і ефективним, тому система може швидко виявити загрози, що є особливо важливим для реальних систем, де час реакції має критичне значення. Наприклад, в автомобільних мережах CAN-bus швидка реакція може бути необхідною для забезпечення безпеки.

Проте є й деякі недоліки. Головним обмеженням є те, що такі системи не можуть виявляти нові або змінені атаки, яких немає в базі даних сигнатур. Тобто, якщо атака нова або модифікована, сигнатурна система не зможе її виявити, оскільки вона не матиме відповідного шаблону в базі даних. Однак це можна виправити, регулярно оновлюючи базу сигнатур. Оновлення дозволяє системі залишатися актуальною та адаптуватися до нових загроз.

Один з можливих способів покращити ефективність сигнатурних систем - це поєднання їх з іншими методами виявлення, такими як виявлення аномалій або технології на основі глибокого навчання. Це дозволить системі виявляти нові загрози, навіть якщо вони ще не були зафіксовані в базі сигнатур. Комбінація таких методів дає змогу підвищити точність виявлення і зменшити кількість хибних спрацьовувань, оскільки система буде не лише порівнювати пакети з

шаблонами, але й аналізувати їхню поведінку.

Наприклад, якщо атака використовує незвичайний шаблон, який не є в базі, методи глибокого навчання можуть допомогти розпізнати таку аномалію за її поведінковими характеристиками. Така комбінація методів дозволяє створити більш ефективну і гнучку систему виявлення вторгнень, яка здатна адаптуватися до нових загроз.

Таким чином, методи на основі сигнатур - це швидкий і точний спосіб виявлення відомих атак. Проте для їх підвищення ефективності необхідно регулярно оновлювати базу даних сигнатур і комбінувати їх з іншими методами, такими як виявлення аномалій або глибоке навчання, щоб система могла виявляти нові або модифіковані загрози [25].

Ще одним з методів є Anomaly-based IDS - це метод який має ключову роль у забезпеченні безпеки в сучасних мережах, включаючи CAN-bus. Цей метод дозволяє виявляти нові або модифіковані атаки, які ще не були зафіксовані в базах даних сигнатур. Це робить систему більш адаптивною і здатною захистити від невідомих загроз.

Даний метод виявлення аномалій на відміну від сигнатурних систем, не базуються на заздалегідь відомих шаблонах атак, а виявляють аномалії в мережевому трафіку. Це дає їм змогу реагувати на нові або незнайомі загрози. Наприклад, нові види атак можуть використовувати нестандартні техніки або модифікації вже відомих методів, що не дозволяє сигнатурним системам виявити такі загрози. Метод виявлення мережевих аномалій та потенційних загроз дозволяє виявляти такі аномалії, навіть якщо вони ще не були зафіксовані в базах даних.

Оновлення бази даних сигнатур є великим завданням для систем, які використовують методи на основі сигнатур. Для підтримки ефективності системи необхідне регулярне оновлення сигнатур, що потребує значних витрат часу та ресурсів. У випадку методу виявлення мережевих аномалій це не є такою проблемою, оскільки вони не потребують регулярного оновлення шаблонів атак. Вони здатні адаптуватися до змін у мережевому середовищі, оскільки виявляють

не конкретні загрози, а будь-які відхилення від нормального трафіку [26].

Даний метод виявлення атак є досить гнучкими. Вони здатні працювати з будь-яким типом атак, оскільки вони не прив’язані до конкретних шаблонів. Це дозволяє їм працювати в різноманітних середовищах і на різних мережах, у тому числі в складних або швидко змінюваних середовищах, таких як мережі CAN-bus в автомобілях. Зміни в мережевій інфраструктурі або поведінці користувачів не обов'язково призведуть до зниження ефективності методу, на відміну від сигнатурних, які потребують постійного оновлення шаблонів.

Основний недолік методу - це висока ймовірність хибних спрацьовувань. Система може вважати нормальною поведінку, яка є відхиленням від звичайного шаблону, що призводить до зайвих попереджень і перевантаження оператора. Такі ситуації можуть виникати, якщо система не точно визначає, що є "нормальним" для конкретної мережі або інфраструктури, зокрема для специфічних мереж, таких як CAN-bus.

Щоб зменшити кількість хибних спрацьовувань та забезпечити ефективність виявлення загроз, метод потребує ретельного налаштування та навчання моделей на основі нормальної поведінки. Однак, якщо система неправильно визначає норму або її поведінка змінюється, це може призвести до значних проблем з точністю виявлення атак. Навчання системи для правильного розпізнавання нормального трафіку в конкретній мережі є складним, особливо у швидко змінюваних або нових середовищах.

Оскільки даний метод має проблеми з хибними спрацьовуваннями та адаптацією до нових загроз, його можна комбінувати з іншими методами, такими як сигнатурні системи. Це дозволить знизити ймовірність помилкових спрацювань, виявляючи як відомі загрози через сигнатури, так і нові аномалії через методи на основі відхилень [27].

Також є гібридний метод, він поєднує два основні підходи: сигнатурний і виявлення аномалій**,** що дозволяє отримати вищий рівень захисту в порівнянні з використанням тільки одного з них. Сигнатурні методи працюють за принципом порівняння трафіку з відомими шаблонами атак. Вони швидко виявляють

загрози, які вже зафіксовані та мають чіткі ознаки, наприклад, типові шкідливі пакети даних. А ось метод виявлення аномалій відрізняється тим, що він вивчає звичайну поведінку мережі, визначають "нормальний" трафік і виявляють відхилення від цього шаблону. Це дозволяє знаходити нові або модифіковані загрози, яких ще не було в базах даних.

Застосування гібридного методу дає низку переваг. Перша і головна перевага полягає в тому, що система здатна одночасно знаходити як відомі загрози через сигнатури, так і нові, не зафіксовані загрози через аномалії. Це суттєво знижує ймовірність того, що загроза залишиться непоміченою. Крім того, поєднання двох підходів допомагає зменшити кількість хибних спрацьовувань. Якщо, наприклад, одна система спрацьовує на новий тип атаки, інша може її не помітити, але в комплексі ці два підходи взаємно покривають слабкі місця один одного. Гібридні методи також забезпечують велику гнучкість і адаптивність, що особливо важливо для динамічних середовищ, таких як CAN-bus в автомобілях, де може змінюватися структура мережі або з'являтися нові пристрої.

Однак у гібридних методів є і недоліки. По-перше, оскільки система використовує два різні методи виявлення атак, вона вимагає більше обчислювальних ресурсів і пам'яті, що може призвести до високих витрат на технічне забезпечення. По-друге, налаштування гібридної системи та інтеграція її в існуючу інфраструктуру можуть бути складними. Кожен метод потребує окремого налаштування, і їх взаємодія повинна бути бездоганною для ефективної роботи системи. Також необхідно регулярно оновлювати обидва методи: сигнатурний підхід потребує оновлення бази даних для виявлення нових атак. Це потребує додаткових ресурсів і зусиль для підтримки системи в актуальному стані.

Отже, гібридний метод поєднує переваги двох підходів, дозволяючи одночасно виявляти як відомі загрози, так і нові, не зафіксовані в базах даних. Це робить його ефективним інструментом для захисту мереж, таких як CAN-bus, але для його успішного застосування потрібно враховувати додаткові витрати на

ресурси, складність налаштування та потребу в регулярному оновленні [28].

Ще одним з підходів до виявлення атак у мережах CAN-bus є рішення на основі апаратного забезпечення - Hardware-based Solutions. Цей метод передбачає використання спеціалізованого апаратного забезпечення для виявлення атак у мережах, зокрема в мережах CAN-bus. До такого обладнання належать, наприклад, програмовані вентильні матриці або інтегральні схеми спеціального призначення. які дають змогу реалізовувати спеціалізовані алгоритми моніторингу та аналізу мережевого трафіку в режимі реального часу. Принцип роботи таких систем полягає в тому, що апаратне забезпечення здійснює обробку та аналіз трафіку безпосередньо на рівні апаратного компонента, що мінімізує затримки та забезпечує високу швидкість реагування на загрози.

Апаратні системи здійснюють моніторинг трафіку, порівнюючи його з відомими шаблонами атак або визначаючи відхилення від нормальної поведінки мережі. У разі виявлення загрози система оперативно реагує, здійснюючи блокування шкідливих пакетів або припиняючи дію атаки. Однією з головних переваг апаратних рішень є їхня висока швидкість і надійність, оскільки обробка даних відбувається безпосередньо на апаратному рівні, що мінімізує затримки та збільшує стабільність системи, адже апаратне забезпечення не залежить від операційної системи.

Однак, незважаючи на високу ефективність, апаратні рішення мають і недоліки. Вони можуть бути дорогими в порівнянні з програмними системами через необхідність у спеціалізованому обладнанні та його інтеграції в існуючі мережі. Крім того, апаратні рішення менш гнучкі та можуть вимагати фізичного оновлення при адаптації до нових загроз або змін у мережевому середовищі. Масштабування таких систем також може бути складним і дорогим, оскільки для обробки більшого обсягу трафіку потрібне додаткове обладнання. Таким чином, хоча апаратні рішення забезпечують високу швидкість і стабільність, їх вартість і обмежена гнучкість можуть бути бар'єрами для широкого використання в деяких випадках [29].

Ще один ефективний варіант є використання програмних рішень Software-based Solutions для виявлення атак у мережах, зокрема в мережах CAN-bus. Даний варіант використовує стандартне програмне забезпечення, яке працює на загальних комп'ютерних платформах: сервери, ПК або віртуальні машини. Такі системи здатні аналізувати мережевий трафік і виявляти загрози за допомогою алгоритмів, що виконуються на програмному рівні, що забезпечує високу гнучкість і масштабованість. Програмні IDS використовують методи, засновані на сигнатурах атак, аномаліях або машинному навчанні для виявлення шкідливих дій у мережі. Принцип роботи програмних рішень полягає в тому, що вони збирають, обробляють і аналізують дані, що передаються через мережу. Програмне забезпечення може бути налаштоване на моніторинг трафіку, порівняння його з відомими шаблонами атак або на виявлення аномалій, що вказують на підозрілу поведінку в мережі. Після виявлення загрози програма генерує попередження або вживає заходів щодо блокування шкідливого трафіку.

Переваги програмних рішень включають високу гнучкість, здатність до швидкого оновлення та адаптації, зокрема через можливість використання нових алгоритмів, машинного навчання та інтелектуальних систем. Вони можуть працювати на стандартному комп'ютерному обладнанні, що значно знижує витрати на апаратне забезпечення та дозволяє масштабувати систему відповідно до змін у мережевому середовищі. Крім того, програмні рішення можуть бути більш доступними та менш затратними, ніж апаратні, оскільки вони не вимагають спеціалізованого обладнання.

Однак, програмні рішення мають і свої недоліки. Одним із головних є те, що вони можуть бути обмежені у швидкості обробки даних, оскільки процеси аналізу виконуються на загальному програмному рівні, а не на спеціалізованому апаратному обладнанні. Це може призвести до затримок у виявленні атак у мережах з великим обсягом трафіку. Крім того, такі системи можуть бути вразливі до атак, орієнтованих на програмне забезпечення, таких як вразливості в операційних системах або додатках, що використовуються для моніторингу трафіку. Враховуючи ці фактори, програмні рішення можуть вимагати більшого

обсягу обчислювальних ресурсів і бути менш ефективними в порівнянні з апаратними, особливо для великих та швидко змінюваних мереж [30].

Також одним з методів є метод на основі глибокого навчання. Даний метод є одни з найбільш перспективних підходів для виявлення атак у мережі CAN-bus у режимі реального часу. Цей метод використовує складні нейронні мережі для автоматичного навчання на великих обсягах даних, що дозволяє розпізнавати складні шаблони та виявляти нові або раніше невідомі загрози.

Принцип роботи таких систем полягає в тому, що вони аналізують нормальний трафік мережі, вивчають його характеристики та виявляють аномалії, які можуть бути ознакою атаки. Після етапу навчання система може виявляти нові загрози, навіть якщо вони не були зафіксовані в базах даних.

Однією з основних переваг є здатність до виявлення нових загроз, що не були раніше зафіксовані, а також адаптивність до змін у мережевій інфраструктурі, знижуючи кількість хибних спрацьовувань. Однак, цей метод має і кілька недоліків. По-перше, він потребує великих обчислювальних ресурсів, оскільки для навчання нейронних мереж необхідні потужні сервери або графічні процесори. По-друге, для досягнення високої ефективності виявлення необхідно багато даних для навчання, що може бути складно в умовах обмежених джерел інформації. Тривалий процес навчання також може бути проблемою, особливо в реальних умовах, де час реакції важливий. Крім того, глибокі нейронні мережі часто є "чорними ящиками", і може бути складно зрозуміти, чому система класифікує певний трафік як аномальний. Таким чином, хоча методи на основі глибокого навчання мають високий потенціал для виявлення складних атак, їх впровадження вимагає значних ресурсів і правильного налаштування [31].

Для більш детального аналізу ефективності методів виявлення атак у мережах CAN-bus наведена таблиця, яка показує, як кожен метод відповідає на специфічні типи атак, такі як DoS, Bus-off, Spoofing та інші. Це дозволить чітко зрозуміти переваги та обмеження кожного методу в різних умовах таблиця 2.1.

Таблиця 2.1 – Порівняння методів виявлення атак у мережах CAN-bus

Типи атак	Сигнатурний метод	Anomaly-based IDS	Гібридний метод	Апаратне рішення	Програмне рішення	Метод на основі глибокого навчання
Атаки відмови в обслуговуванні	Може виявляти специфічні патерни трафіку	Може виявляти аномалії в трафіку	Комбінація методів дає точні результати	Спеціалізоване обладнання для фільтрації трафіку	Залежить від програмної логіки для фільтрації	Може ефективно виявляти складні атаки за допомогою моделей
Атака "Bus-off"	Важко виявити конкретні патерни	Важко виявити конкретні патерни	Відмінності можуть бути менш очевидними	Дозволяє виявляти непередбачені зміни на рівні апаратного забезпечення	Залежить від можливостей програмного забезпечення	Глибоке навчання може моделювати аномальні сценарії
Freeze Doom Loop атака	Потрібно точне налаштування	Легко виявляється зміна в циклах	Поєднання методів дає більшу точність	Може бути використано для стабільного моніторингу	Програма може швидко реагувати на замороження	Глибоке навчання може виявити за допомогою виявлення патернів
Spoofing атаки	Швидко виявляє підроблені кадри	Легко виявляються аномалії в поведінці	Аналіз комбінації різних типів атак	Чітко фіксуються аномальні кадри	Швидко реагує на змінені дані	Ідеально підходить для виявлення підроблених кадрів
Error Passive Spoofing	Мало специфічних патернів	Легко виявити за допомогою аналізу помилок	Підроблені кадри можуть бути схожі на нормальні	Апаратне рішення дозволяє більш точно відслідковувати помилки	Програмні рішення ефективно відслідковують зміни в кадрах	Глибоке навчання дозволяє виявляти навіть найменші зміни в кадрах
Double Receive Attack	Зазвичай мало визначених патернів	Аномалії можуть бути виявлені при великій кількості отриманих кадрів	Підроблені кадри стають легшими для виявлення за допомогою інших методів	Для великої кількості кадрів може бути обмежена точність	Програмні рішення допомагають виявляти подвійне отримання	Адаптивні атаки можуть бути розпізнані за допомогою великих даних і навчання
Атаки на перехоплення та маніпуляцію кадрами	Чітко визначаються підроблені кадри	Порушує звичайні патерни	Помилки можна легко аналізувати за допомогою кількох	Маніпуляція кадрами може бути зафіксована на рівні обладнання	Програмні рішення допомагають виявляти маніпуляцію кадрами	Може виявляти помилки і аномалії в трафіку

З огляду на таблицю видно що метод на основі глибокого навчання є найбільш перспективним для виявлення атак у мережах CAN-bus. Тому що він дозволяє ефективно виявляти складні атаки, завдяки здатності моделювати

аномальні сценарії, цей метод може виявляти навіть складні підроблені кадри та атаки, які важко виявити традиційними методами. Також він може, виявляти нові атаки без необхідності постійного оновлення сигнатур, що робить його більш гнучким і стійким до змін у поведінці атакуючих. Даний метод має високу точність при великих обсягах даних. Завдяки використанню великих даних і алгоритмів глибокого навчання, цей метод має здатність виявляти навіть найменші зміни в мережевому трафіку, що підвищує його точність та ефективність. Має покращену здатність до обробки аномалій, що можуть бути ознаками атаки, навіть якщо ці аномалії не мають явних патернів або сигнатур.

Проведений аналіз існуючих підходів до захисту мережі CAN-bus був спрямований на виявлення їх сильних і слабких сторін з метою обґрунтованого вибору найбільш ефективного рішення для майбутньої розробки. Сигнатурні методи забезпечують високу точність для відомих атак, але не виявляють нових загроз; методи виявлення аномалій — більш гнучкі, проте схильні до хибних спрацьовувань. Гібридні системи дозволяють поєднати переваги обох підходів, забезпечуючи кращу ефективність в умовах обмежених можливостей CAN-протоколу. З огляду на це, доцільним є вибір гібридного підходу з можливим залученням алгоритмів глибокого навчання, що дозволить підвищити точність і адаптивність системи виявлення атак у мережах CAN-bus.

2.3 Метод реалізації виявлення вторгнень

Метод реалізації виявлення вторгнень у системі захисту мережі CAN-bus базується на комплексному підході, який спирається на аномальний аналіз трафіку з використанням евристичних правил та базової статистичної оцінки параметрів мережевих повідомлень. Це гібридне рішення, яке поєднує сигнатурний підхід із виявленням аномалій, забезпечуючи ширше охоплення типів атак. Головною метою цього методу є можливість ідентифікувати як відомі типи атак, що мають характерні ознаки, так і потенційно нові, раніше не

зафіксовані загрози. Для досягнення цієї мети використовується принцип побудови поведінкових моделей нормальної роботи мережі, які служать еталоном для подальшого порівняння вхідного трафіку.

Першим і дуже важливим кроком у цьому процесі є навчання моделі на еталонному трафіку, що відповідає нормальному, неаномальному функціонуванню мережі. Під час цього етапу збирається детальна статистика, яка включає інформацію про допустимі інтервали часу між послідовними повідомленнями певного CAN ID, кількість унікальних ідентифікаторів, що зустрічаються в мережі, а також частоту появи кожного з цих ідентифікаторів. Крім того, аналізується характер зміни даних у полі “data” кожного пакету, що BUS-OFF ATTACK дозволяє сформувати профіль типової динаміки значень. Вся ця інформація зберігається у вигляді референтного профілю - свого роду «образу» нормального функціонування мережі, який виступає еталоном при подальшому порівнянні**.** Сформовані характеристики звичайного трафіку лягають в основу системи виявлення вторгнень, забезпечуючи можливість порівняння поточної активності з раніше визначеною нормою. На рисунку 2.2 наведено узагальнену схему алгоритму виявлення вторгнень у CAN-bus мережі.

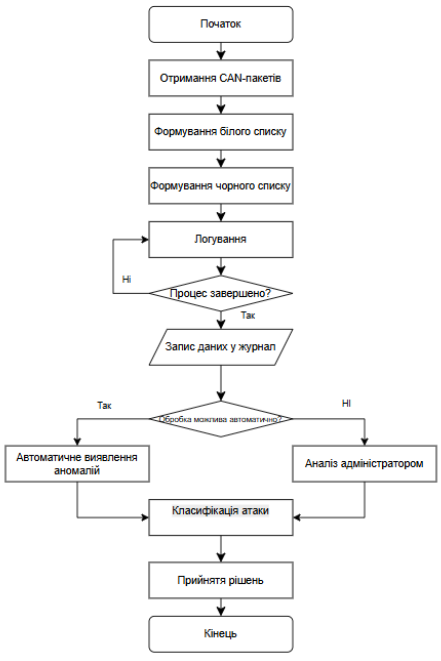


Рисунок 2.2 - Алгоритм виявлення вторгнень у CAN-bus мережі

Основним компонентом алгоритму є модуль порівняння отриманих CAN-пакетів із сформованим еталоном. Цей модуль реалізує багаторівневу перевірку параметрів повідомлень. Зокрема, він контролює частоту появи кадрів із певним CAN ID. У випадку, якщо кількість таких кадрів за одиницю часу значно перевищує встановлені пороги, система фіксує цю подію як потенційну атаку типу Flooding, яка спрямована на перевантаження мережі численними однаковими повідомленнями. Далі здійснюється ідентифікація спроб дублювання або підробки кадрів, що може свідчити про атаки типу spoofing або replay. Якщо система виявляє кілька повідомлень з однаковим ID, але різними даними або незвичною періодичністю їх появи, це підвищує рівень підозри та викликає відповідне сповіщення. Окрім цього, у модулі передбачена поведінкова оцінка зміни значень у полі даних. Якщо дані змінюються поза межами фізично можливих або очікуваних значень, або при цьому виявляються атипові градієнти змін, це вважається індикатором маніпуляції інформацією.

Для зменшення кількості хибнопозитивних спрацьовувань у систему введено багатоступеневу логіку фільтрації повідомлень. Перший рівень аналізує кожен кадр індивідуально, після чого виконується кумулятивна оцінка подій за часовими вікнами, наприклад, тривалістю 1 секунду. Такий підхід дозволяє системі виявляти як короткочасні ін’єкції атакувального трафіку, так і довготривалі аномалії, які можуть вказувати на системні проблеми або тривалі атаки. Це забезпечує високий рівень надійності у визначенні загроз і мінімізує ризик помилкових спрацьовувань, що дуже важливо для критичних застосувань у автомобільній електроніці.

Особливістю даного методу є його гнучкість і адаптивність, досягнуті завдяки використанню набірних правил, які можуть бути легко змінені або доповнені без необхідності вносити суттєві зміни в архітектуру програми. Наприклад, можна додати нові правила, що враховують частоту появи кадрів з певним ID, або включити реагування на незвичні комбінації довжин кадрів, які раніше не були враховані. Такий підхід дозволяє швидко адаптувати систему під специфіку конкретної мережі або під нові типи атак, які можуть з’явитися з

часом.

Важливо зазначити, що метод не покладається на складні алгоритми машинного навчання, які часто потребують значних обчислювальних ресурсів і мають низьку прозорість у прийнятті рішень. Замість цього, використовується детермінований контроль властивостей трафіку, що забезпечує високу швидкодію, простоту налаштування і можливість чіткого пояснення причин спрацьовування системи. Це робить метод придатним для вбудованих систем із обмеженими обчислювальними ресурсами, які часто застосовуються у транспортних засобах.

Таким чином, реалізований метод виявлення атак у мережі CAN-bus забезпечує ефективний захист від широкого спектру загроз, поєднуючи простоту, гнучкість і високу надійність, що дозволяє інтегрувати його у різні апаратні платформи та адаптувати під конкретні умови експлуатації мережі.

2.4 Вибір інструментів та середовища розробки

Для реалізації системи виявлення вторгнень у мережі CAN-bus у режимі реального часу було обрано набір програмних інструментів, який відповідає вимогам до функціональності, сумісності з операційною системою Windows та можливості моделювання трафіку без потреби у фізичному підключенні до CAN-bus.

Операційною системою, що використовувалася у межах проєкту, стала Windows 10. Це стабільне та поширене середовище, яке забезпечує зручність у роботі з Python, підтримку віртуального середовища, а також сумісність з обраними бібліотеками. З практичного боку, дана ОС є типовою для більшості персональних комп’ютерів, що спрощує розгортання та тестування проєкту.

Основною мовою програмування було обрано Python версії 3.10. Цей вибір зумовлений її простою синтаксичною структурою, великою кількістю готових бібліотек для обробки даних та роботи з мережевими протоколами, а також

активною підтримкою спільноти. Python добре підходить для побудови прототипів та реалізації алгоритмів виявлення аномалій і атак у мережах [32].

У якості середовища розробки застосовувалася PyCharm Community Edition. Ця IDE забезпечує зручну навігацію по проєкту, можливість роботи з віртуальним середовищем, налаштування структури директорій, а також ефективне відлагодження коду. У межах проєкту було створено окреме середовище з встановленими необхідними пакетами, що дозволило уникнути конфліктів версій [33].

На етапі попередньої реалізації передбачалося використання бібліотеки Scapy, зокрема її додаткового модуля для роботи з CAN-пакетами. Проте під час імпорту модуля scapy.contrib.can виникла критична помилка, пов’язана з відсутністю його підтримки у середовищі Windows. Таким чином, робота з CAN-пакетами за допомогою Scapy виявилася неможливою.[34]

Як альтернативу було вирішено реалізувати власний інструмент моделювання CAN-трафіку шляхом створення спеціального класу для представлення CAN-повідомлень. Такий підхід забезпечив повну гнучкість у налаштуванні структури повідомлень, ідентифікаторів та даних, що дозволило моделювати поведінку реальної CAN-мережі, не покладаючись на зовнішні обмеження бібліотек.

Використання власної моделі повідомлень дозволило точно контролювати формат, структуру та логіку передачі пакетів, а також забезпечило сумісність з іншими модулями проєкту, включаючи генерацію атак та механізми виявлення аномалій. Такий підхід виявився найбільш оптимальним з урахуванням поставлених вимог, технічних обмежень середовища та цілей дослідження.

2.5 Схема роботи

Практична частина роботи полягає у створенні віртуального середовища для моделювання роботи системи виявлення вторгнень у мережі CAN-bus та

реалізації типових атак, з якими може стикатися така мережа. Робота над реалізацією розпочинається з налаштування робочого середовища: операційною системою виступить Windows 10, середовищем розробки - PyCharm,а мовою програмування буде Python, головним інструментом для генерації, модифікації та аналізу мережевого трафіку стане бібліотека Scapy. Зважаючи на відсутність повноцінної фізичної мережі CAN-bus, реалізувати симуляцію мережі через створення класів, що емулюють структуру CAN-пакетів, зокрема таких елементів як ідентифікатор, поле даних та довжина повідомлення [35-36].

Основна увага буде зосереджена на практичному відтворенні трьох найпоширеніших типів атак: Replay, Flooding та Spoofing. Для реалізації Replay-атаки буде створено клас CANMessage, який містить поля ідентифікатора та масиву байтів як вмісту повідомлення, а також метод відображення вмісту у зрозумілому форматі. Симуляція Replay-атаки здійснюватиметься шляхом багаторазового надсилання одного й того ж CAN-повідомлення з фіксованим інтервалом, що дозволить імітувати повторну передачу легітимного пакету типовий прийом зловмисника для викликання небажаної реакції елементів транспортного засобу. Далі буде реалізована Flooding-атака, яка має на меті перенавантажити шину великою кількістю беззмістовних пакетів. Для цього використовуватиметься цикл генерації випадкових повідомлень з випадковими ідентифікаторами та полем даних довжиною 8 байт, що відповідає стандартному CAN-повідомленню. У рамках цієї атаки буде надіслано десятки повідомлень у дуже короткий проміжок часу, що дозволило перевірити, чи зможе мережа обробити великий обсяг трафіку без помилок і як відреагує система виявлення вторгнень. Третій тип атаки Spoofing буде реалізований шляхом генерації підроблених пакетів з легітимними ідентифікаторами, але зі зміненим полем даних, що дозволить симулювати підміну справжнього пристрою зловмисником. У цьому випадку важливо продемонструвати здатність IDS-системи виявляти аномалії у поведінці пристроїв навіть у разі правильної структури та ID повідомлень.

Створення та відправлення таких пакетів здійснюватиметься із

використанням функціоналу бібліотеки Scapy. Усі ці типи атак будуть реалізовані як окремі Python-скрипти, що дозволить запускати їх незалежно один від одного та створювати зручне середовище для покрокового тестування. Кожна з атак супроводжуватиметься логами з відображенням ID, довжини та вмісту даних у шістнадцятковому форматі, що дозволить відстежити, як змінюється структура CAN-повідомлень при реалізації атаки. Результати запуску скриптів виводимуться у консоль, що також є важливим з точки зору фіксації результатів для подальшого аналізу. У результаті практичної реалізації симуляційного середовища та тестових атак буде отримано інформацію про поведінку системи у відповідь на зловмисні дії. Такий підхід дасть змогу не лише імітувати реальні загрози, а й оцінити ефективність підходів до виявлення вторгнень на основі аномалій.

На підставі змодельованого трафіку можливо у майбутньому реалізувати більш складні механізми аналізу, зокрема машинне навчання, для точного виявлення невідомих або комбінованих типів атак. Уся ця схема роботи буде демонструвати ефективність симуляційного підходу для дослідження систем інформаційної безпеки транспортних мереж та слугуватиме надійною основою для розробки повноцінних рішень з виявлення вторгнень у реальному часі в мережах типу CAN-bus [37].

2.6 Висновки до розділу

У цьому розділі розглянуто методологію побудови системи виявлення вторгнень у мережі CAN-bus, а також обґрунтовано вибір архітектури, методів та інструментів, застосовуються у роботі. Розроблена модель буде симуляційною - це означає, робота буде не з фізичною CAN- bus, а з її програмною імітацією, створеною на мові Python. Такий підхід дозволить мені зберегти гнучкість у розробці, а також уникнути потреби у дорогому спеціалізованому обладнанні. Використання віртуального середовища дасть змогу глибоко проаналізувати

роботу CAN-мережі в умовах, що наближені до реальних, а також відтворити типові сценарії атак без ризику пошкодження реального устаткування.

Використовуючи об'єктно-орієнтовану модель CAN-повідомлення, реалізувати функціонал, який дозволить мені створювати, змінювати та аналізувати структуру CAN-пакетів. Що дасть мені змогу перейти до реалізації конкретних атак, які становлять найбільшу загрозу для CAN-мереж. У межах даної моделі мною будуть змодельовані три базові типи атак: Replay, Flooding та Spoofing. Кожна з них репрезентує свій підхід до зловмисного втручання: від дублювання законних повідомлень до масової генерації трафіку або підміни інформації зі збереженням легітимного ідентифікатора. Така різноманітність дозволить протестувати реакцію IDS-системи в різних умовах і виявити, які саме характеристики повідомлень можуть сигналізувати про порушення.

Зокрема, Replay-атака стане чудовим прикладом використання вже перехопленого повідомлення для перевірки того, чи реагує система на дублікати, які хоч і формально коректні, але не відповідають очікуваному патерну роботи пристрою. Flooding-атака, у свою чергою, дозволить протестувати пропускну здатність віртуальної шини і виявити, наскільки стійкою є система до перевантаження беззмістовними повідомленнями. А Spoofing-атака акцентуватиме увагу на важливості не лише структури, а й семантики переданих даних, адже підроблене повідомлення може виглядати зовсім легітимно - проте містити небезпечну або хибну інформацію.

Реалізація кожного з типів атак у вигляді окремого Python-скрипта дозволить мені створити гнучку, модульну систему, де будь-яку атаку можна окремо запустити, проаналізувати та модифікувати. Завдяки цьому отримаємо можливість зручно досліджувати реакцію системи на кожен окремий вектор загроз, виводити у консоль детальну інформацію про передані повідомлення, а також логувати результати, що є надзвичайно важливим для подальшого аналізу. Це також створить передумови для інтеграції таких логів у майбутньому в більш складні системи моніторингу та аналізу.

Особливу увагу буде приділено не лише технічному створенню симуляцій,

а й правильній організації структури моделі. Передбачено розмежування між формуванням повідомлень, їх відправкою та відображенням результатів, що відповідає принципам чистої архітектури та дозволить легко масштабувати систему. Наприклад, у майбутньому можна буде додати ще складніші типи атак, зокрема модифікацію трафіку, або реалізувати базові алгоритми машинного навчання для автоматичного виявлення аномалій.

Такий підхід забезпечує гнучкість у розробці, а також дозволяє уникнути потреби у використанні дорогого спеціалізованого обладнання. Віртуальне середовище дає змогу моделювати типові сценарії атак, аналізувати реакцію системи на загрози, не ризикуючи пошкодити реальне устаткування.

На основі отриманих результатів буде можливим не лише перевірити працездатність системи виявлення атак, але й провести повноцінне тестування її надійності в умовах, наближених до реального середовища експлуатації. Це дозволить виявити слабкі місця у механізмах реагування, скоригувати правила виявлення аномалій, а також протестувати поведінку системи в разі складних комбінованих атак. Аналіз реакції на кожен тип загрози допоможе уточнити параметри виявлення, встановити оптимальні порогові значення та зменшити кількість хибнопозитивних спрацьовувань.

У перспективі така симуляційна модель може стати основою для побудови повноцінної системи кіберзахисту транспортних засобів. Зокрема, вона може бути розширена за рахунок впровадження механізмів автоматичного оновлення правил на основі зібраних логів, підключення до реального CAN-інтерфейсу для тестування на фізичному рівні, або інтеграції з іншими модулями захисту — такими як моніторинг доступу до шин даних чи оцінка аномальної поведінки електронних блоків управління (ECU). Таким чином, запропонований підхід демонструє високу практичну цінність і здатність до масштабування відповідно до зростаючих вимог у сфері безпеки автомобільних мереж.

3 ВПРОВАДЖЕННЯ ТА ОЦІНКА СИСТЕМИ

3.1 Тестове середовище

Для реалізації та перевірки роботи системи виявлення атак у мережі CAN-bus розроблено спеціальне тестове середовище, яке працює на операційній системі Windows. Це дозволяє мені створити програмну модель CAN-мережі без потреби в реальному фізичному обладнанні. Оскільки доступу до спеціального апаратного забезпечення для підключення до CAN- bus немає, було вирішено моделювати трафік через програмні інструменти, що значно спрощує процес тестування та аналізу.

У якості мови програмування для реалізації системи виявлення атак було обрано Python. Це один із найпопулярніших мов програмування для розробки тестових середовищ завдяки своїй простоті та зручності для швидкої реалізації алгоритмів. Більше того, Python має багатий набір бібліотек, які дозволяють працювати з мережами і підтримують різні формати даних, що робить його ідеальним вибором для моделювання поведінки мережі CAN.

Для розробки використовується PyCharm, оскільки це середовище розробки має зручний інтерфейс, інтегрований термінал, а також підтримує автоматичне підсвічування синтаксису та автодоповнення, що спрощує процес кодування. Початок розробки є реалізації тестового середовища, яке дозволяє створювати фіктивні CAN-пакети та симулювати їхнє відправлення через мережу [38-39].

Спроба використати бібліотеку Scapy для роботи з CAN-пакетами виявилася невдолою, оскільки вона має велику кількість інструментів для маніпуляцій з мережею. Тому що, при спробі реалізувати цей варіант на операційній системі Windows виникає проблема з модулем scapy.contrib.can, який не підтримується на Windows, отримуємо помилку ModuleNotFoundError: No module named 'scapy.contrib.can' [40-41].

Методом вирішенням цієї проблеми є розробити власний клас для моделювання CAN-пакетів у Python. Це дозволить мати більше контролю над

тим, як виглядатимуть пакети та як вони будуть відправлятися. Створивши клас CANMessage, який дозволяє зберігати ідентифікатор пакету та його дані. Клас також містить метод show, який виводить в консоль структуровану інформацію про CAN-пакет. Таким чином, можна змоделювати формат повідомлення, який використовується в реальній CAN-мережі.

Перш за все потрібно створити новий проєкт у PyCharm з назвою CAN-Simulation, після цьог файл can_simulator.py, у якому прописала клас CANMessage. Цей клас дозволяє створювати об’єкти, які містять ідентифікатор повідомлення, дані та можуть виводити інформацію про себе у консоль. На рисунку 3.1 зображено перший симульований CAN-пакет, який буде використовуватись у подальшій моделі для тестування різних типів атак.

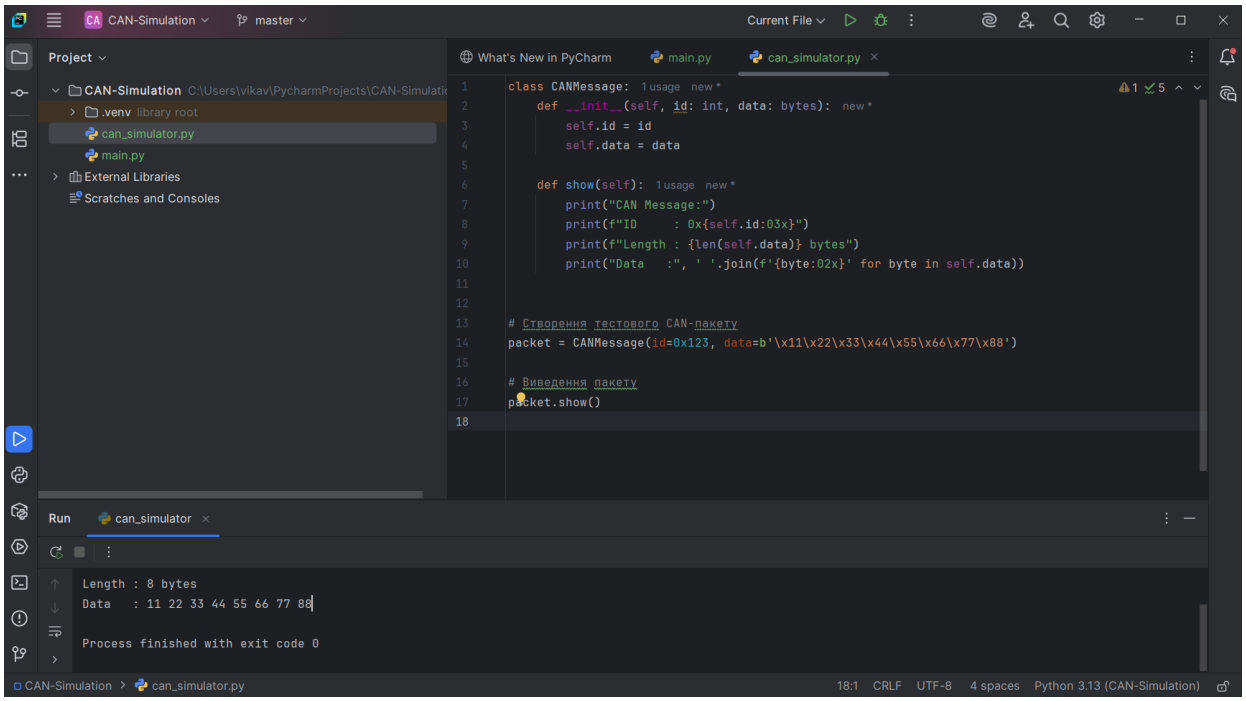


Рисунок 3.1 - Симульований CAN-пакет

Після запуску симулятора було створено фіктивне CAN-повідомлення, яке імітує кадр, переданий у реальній CAN-мережі. У виведеному результаті бачимо структуровану інформацію про створений пакет:

– ID: 0x123 - це унікальний ідентифікатор повідомлення, який у CAN-протоколі визначає пріоритет повідомлення. Чим менше значення ID, тим вищий

пріоритет пакета на шині. У нашому прикладі використано ідентифікатор 0x123 (у шістнадцятковій системі числення), що є умовним значенням для демонстрації.

– Length: 8 bytes - довжина поля даних, тобто скільки байтів інформації передається в цьому пакеті. Стандартний формат CAN дозволяє передавати до 8 байтів у кожному повідомленні. У цьому випадку передається максимально допустиму кількість байтів.

– Data: 11 22 33 44 55 66 77 88 - це власне корисні дані, які передаються у повідомленні. Вони подані у вигляді байтової послідовності. Ці дані можуть представляти будь-яку інформацію: наприклад, швидкість транспортного засобу, положення педалі газу, або інші сенсорні дані.

Таким чином, цей приклад дозволяє на базовому рівні зрозуміти структуру CAN-повідомлення та принцип його симуляції у штучному середовищі без фізичного доступу до шини.

Для реалізації тестового середовища вирішено додати випадкові затримки між відправленнями CAN-пакетів, щоб моделювати реальні умови роботи мережі. Це дозволить симулювати роботу з реальним CAN-трафіком, де пакети не передаються миттєво, а мають певні затримки, зокрема через інтерференцію або обмеження апаратного забезпечення.

Далі створено функцію simulate_delay(), яка генерує випадкові затримки між 10 і 100 мілісекундами. Ця функція дозволяє моделювати умови, схожі на реальну передачу пакетів у CAN-мережі, де пакети не передаються миттєво, а мають певні затримки через різні фактори, такі як інтерференція або обмеження апаратного забезпечення. Результатом роботи цієї функції є затримка, яка додається перед відправленням кожного CAN-пакету. На рисунку 3.2 наведено результат виконання програми, де видно як працює ця функція разом із симульованим CAN-пакетом.

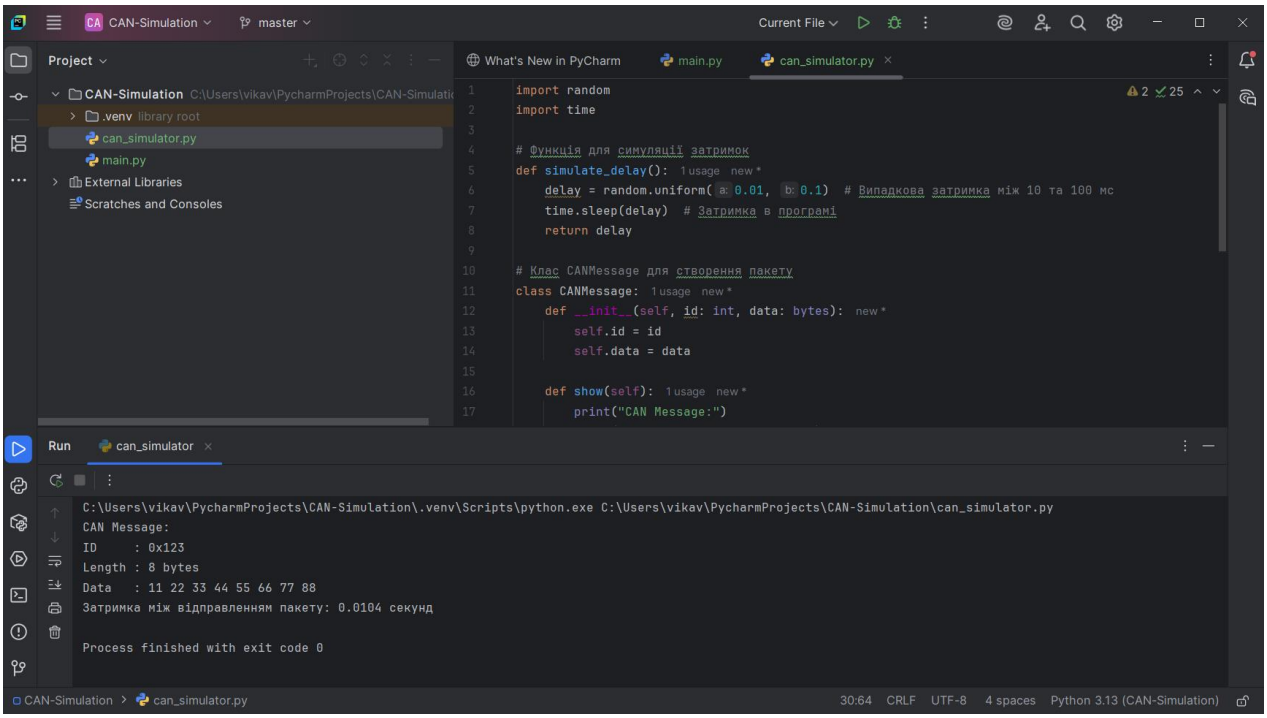


Рисунок 3.2 - Результат виконання програми з функцією simulate_delay()

Після запуску симулятора, отримуємо фіктивне повідомлення, яке виглядає як кадр, переданий у реальній CAN-мережі. Крім того, спостерігається випадкові затримки між відправленнями, що додає ще більше реалістичності в мою модель.

Далі, для повної симуляції, додаємо функціональність для створення кількох повідомлень, їх передачу по "мережі" та подальший аналіз. Однак на даному етапі тестового середовища цей підхід вже дозволяє мені перевірити основні принципи роботи з CAN-пакетами і протестувати функціонування системи виявлення атак.

Використовуючи це тестове середовище, можемо реалізувати різноманітні атаки та перевіряти, як система виявлення реагує на аномалії у трафіку. Це допоможе створити більш ефективні алгоритми для виявлення атак у реальних мережах CAN-bus.

3.2 Тестування системи виявлення вторгнень у мережі CAN-bus

Мета тестування системи виявлення вторгнень у мережі CAN-bus полягає в перевірці її ефективності при виявленні аномалій та атак, що моделюються на основі симульованих CAN-пакетів. У даному випадку для тестування були використані три основні типи атак: Replay, Flooding і Spoofing. Це дозволяє оцінити, як система реагує на різні види вторгнень і виявляє їх у реальному часі.

Перед початком тестування необхідно налаштувати тестове середовище так, щоб воно відповідало умовам реальної роботи CAN-мережі. Як зазначалося в розділі 3.1, для симуляції трафіку на CAN-bus використовуємо Python і бібліотеку PyCharm. Тестування буде проводитися за допомогою спеціально розроблених функцій, які дозволяють генерувати фіктивні CAN-пакети та моделювати затримки, що характерні для реальних умов передачі даних у мережах CAN.

Після створення тестового середовища наступним етапом є реалізація та перевірка роботи системи виявлення атак у моделі мережі CAN-bus. Метою тестування є перевірка, як система реагує на різні типи атак, зокрема Replay-атаку, Flooding-атаку та Spoofing-атаку, які є найбільш поширеними в автомобільних мережах.

Оскільки тестування виконується у віртуальному середовищі, побудованому на мові Python, розроблено серію скриптів, які моделюють кожен тип атаки окремо. Це дозволяє поетапно дослідити поведінку системи та оцінити її здатність розпізнавати аномалії у трафіку.

Першою була реалізована Replay-атака - одна з базових атак у CAN-мережах, яка полягає у повторній передачі раніше перехопленого пакету. Це дозволяє атакувальнику навмисно вводити в систему хибні сигнали, що можуть призвести до помилкових дій на рівні ECU.

Для моделювання цієї атаки використовуємо створений раніше клас CANMessage та додала функцію, яка багаторазово відправляє один і той самий пакет із короткою затримкою між передачами. Повторна поява ідентичних

пакетів у короткий проміжок часу є типовим індикатором Replay-атаки. На рисунку 3.3 зображено роботу Replay-атаки.

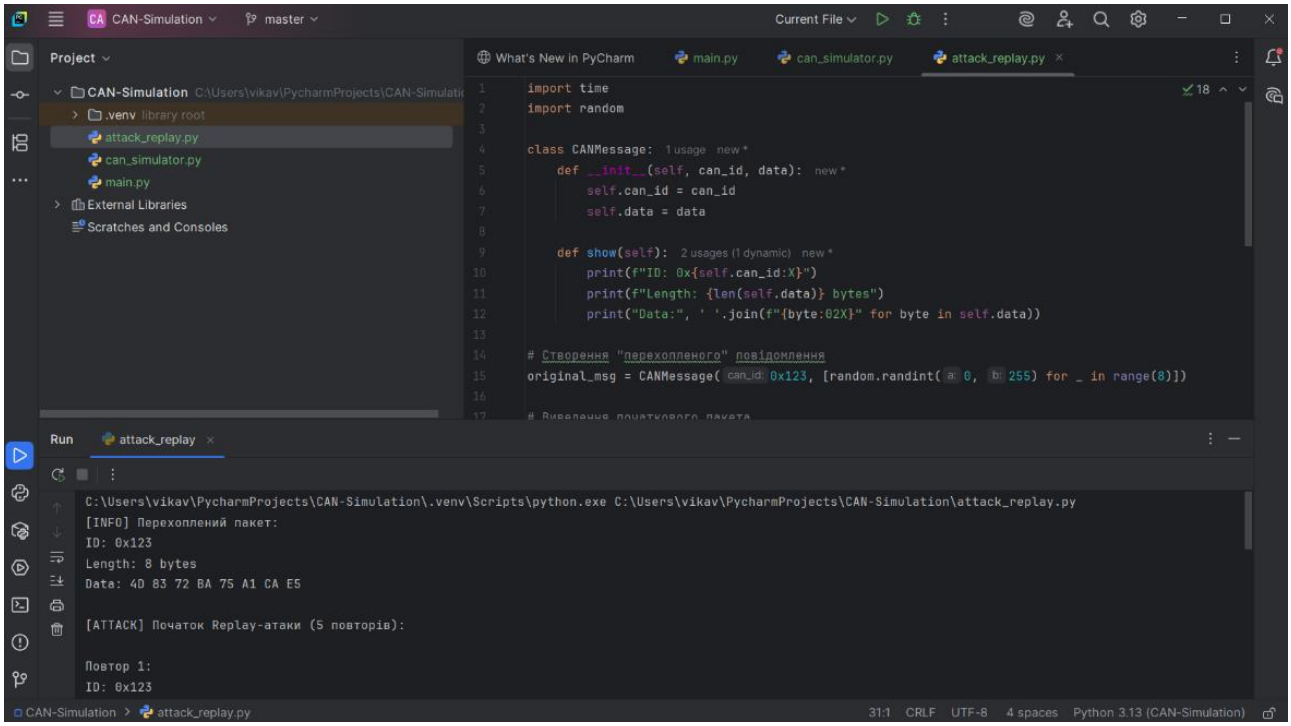


Рисунок 3.3 - Replay-атака

Після успішного запуску симуляції Replay-атаки система згенерувала п’ять ідентичних CAN-повідомлень, що імітують повторну передачу перехопленого пакета. Це дозволяє відтворити типовий сценарій атаки повторного відтворення, під час якої зловмисник намагається ввести систему в оману, багаторазово передаючи вже раніше захоплене повідомлення. Така атака може спричинити небажану поведінку керованих елементів (наприклад, повторне відкриття замка або запуск двигуна), якщо отримувач не перевіряє час або контекст отриманого повідомлення. У даному випадку результат запуску атаки виглядає наступним чином. На рисунку 3.4 зображено результат Replay-атаки.

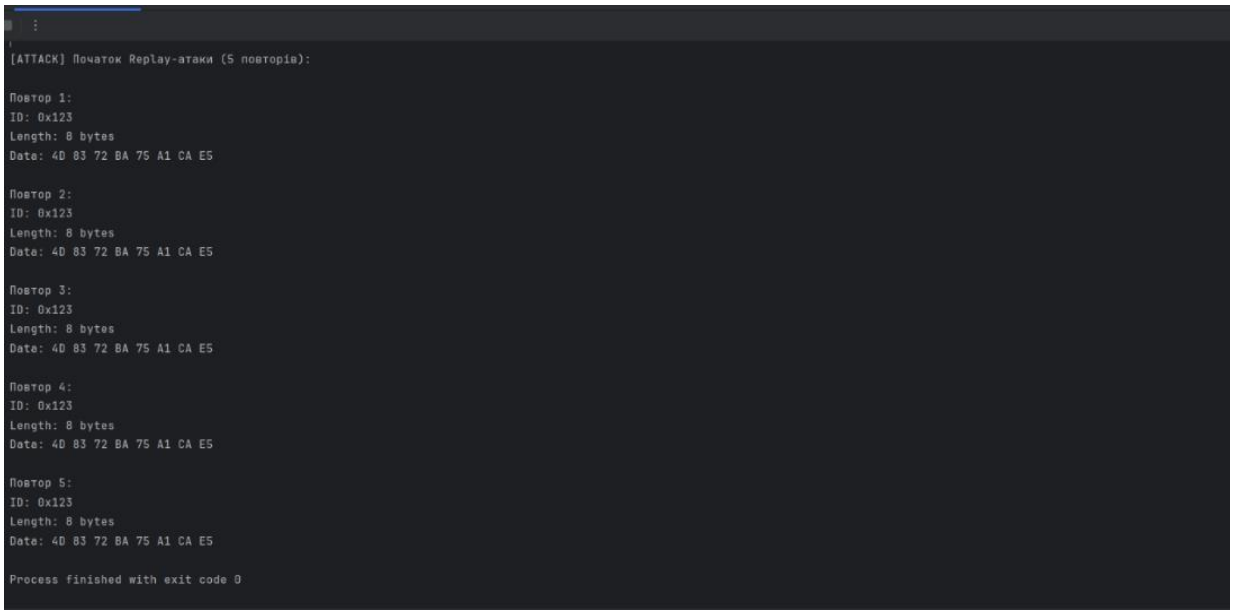


Рисунок 3.4 - Результат Replay-атаки

Наступним типом атак, що був змодельований, є Flooding-атака. Цей тип атаки передбачає надмірне навантаження на шину шляхом швидкої генерації великої кількості повідомлень. Основною метою якого є перевантажити шину, що може призвести до відмови у роботі окремих вузлів або до зниження продуктивності мережі загалом.

Для реалізації даної атаки створюємо цикл, який генерує випадкові CAN-пакети з різними ідентифікаторами та даними, без значних затримок між передачами. Це дозволяє симулювати умови, за яких контролер може не встигати обробляти вхідні повідомлення, ідентифікуючи таку ситуацію як потенційно шкідливу. На рисунку 3.5 зображено роботу Flooding-атаки.

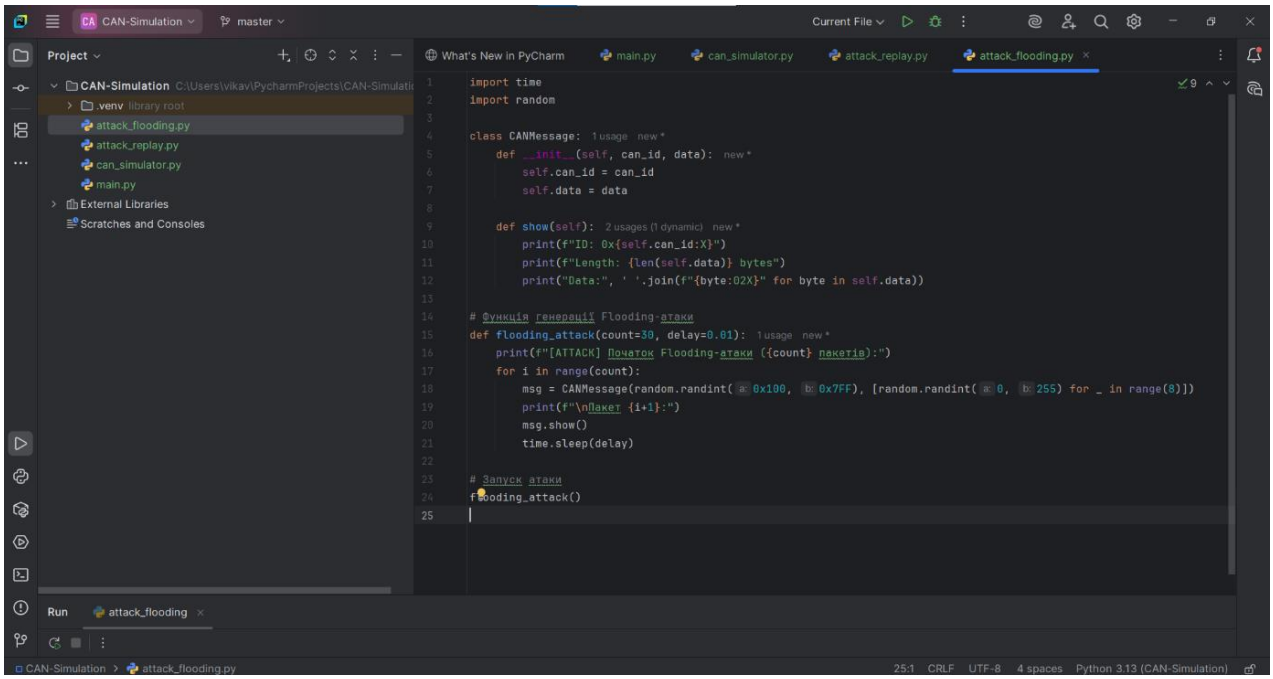


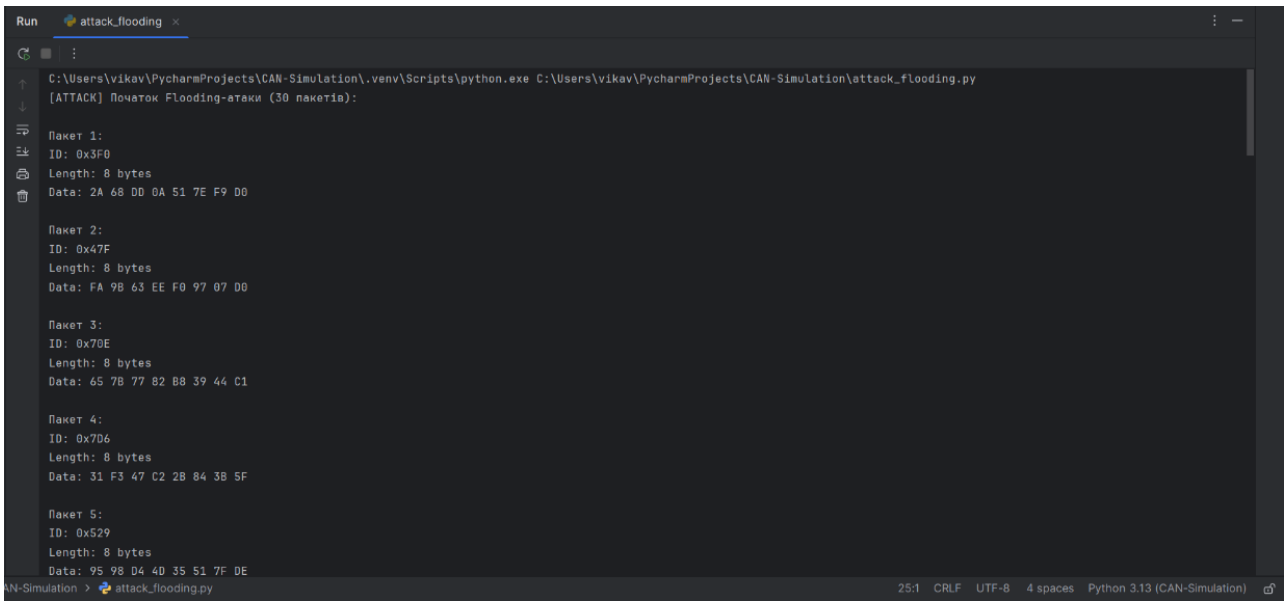
Рисунок 3.5 - Робота Flooding-атаки

У ході тестування Flooding-атаки було виявлено, що система, яка приймає пакети на CAN- bus, зазнає значного перевантаження. Ось детальний аналіз результатів:

– висока кількість пакетів, кожен пакет, який генерується при атаці, має випадкові значення для ідентифікаторів та даних. Завдяки цьому значно збільшується кількість повідомлень на шині, що призводить до перевантаження. У даномувипадку було відправлено 30 пакетів, але на практиці кількість може бути набагато більшою, що призводить до істотних затримок в обробці та передачі даних.

– зниження продуктивності, через таку кількість повідомлень мережа може не встигати обробляти трафік, що веде до затримок у передачі важливих повідомлень. Якщо система має обмежену пропускну здатність, збільшення трафіку може призвести до втрати даних.

– відсутність відмови системи, на відміну від більш агресивних атак, таких як Bus-off, Flooding-атака не обов'язково призводить до повної відмови системи, однак вона може значно знизити її ефективність і швидкість обробки пакетів. На рисунку 3.6 зображено результат роботи Flooding-атаки.



Риснуок 3.6 - Результат роботи Flooding-атаки

Spoofing-атака передбачає підробку CAN-пакетів з фальшивими даними або змінені ідентифікатори (ID) для введення в оману мережу. Такі атаки можуть бути використані з метою саботажу або зловмисного контролю над системою. Наприклад, злочинець може відправити фальшиве повідомлення із зазначенням правильного ID, що змусить систему виконати певні команди.

У цій роботі було реалізовано просту Spoofing-атаку за допомогою Python, використовуючи бібліотеку Scapy для моделювання фальшивих CAN-пакетів. Суть атаки полягає у створенні та відправленні підроблених CAN-пакетів у мережу.

Для реалізації Spoofing-атаки було написано Python-скрипт, який генерує фальшиві пакети та відправляє їх у CAN-мережу. Для цього використовуємо бібліотеку Scapy, яка дозволяє створювати пакети з кастомізованими параметрами та відправляти їх через мережу. На рисунку 3.7 зображено роботу Spoofing-атаки.

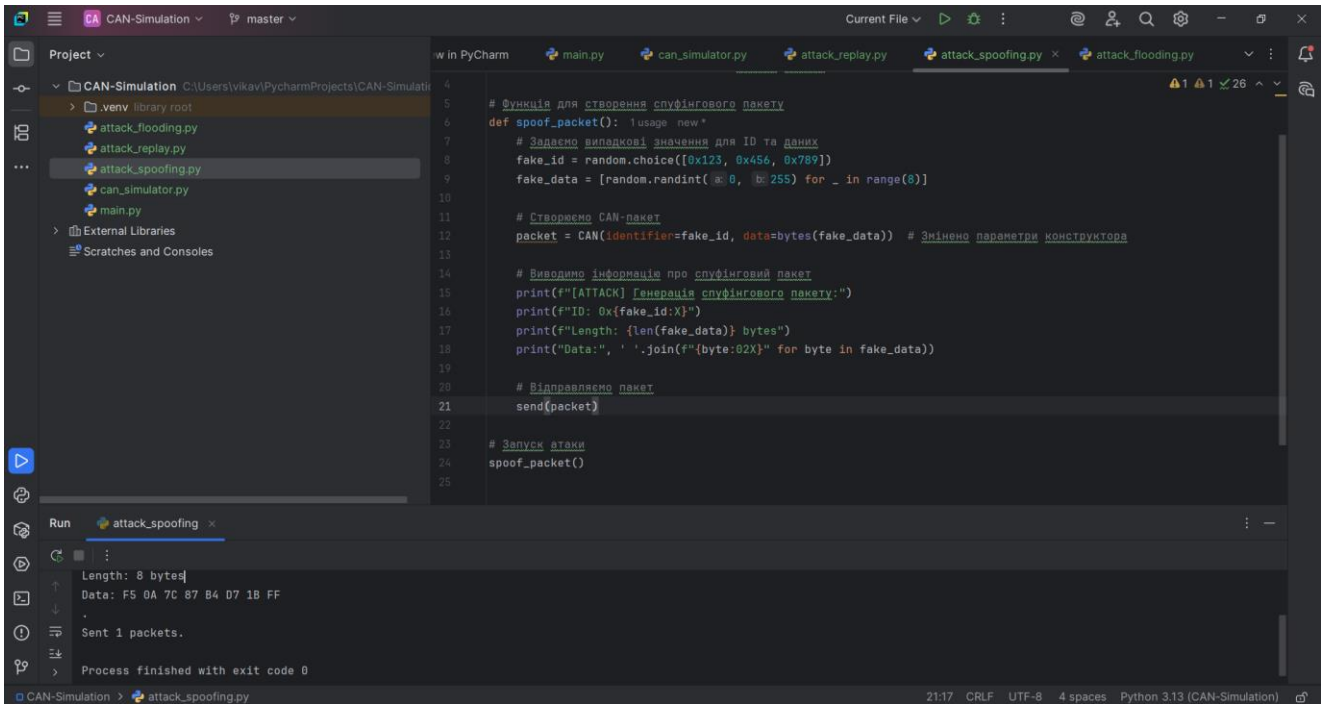


Рисунок 3.7 - Робота Spoofing-атаки

Результат:

– WARNING: MAC address to reach destination not found. Using broadcast - це попередження, яке вказує, що Scapy не змогла знайти конкретну MAC-адресу для відправлення пакету. Тому він використовує широкомовну трансляцію (broadcast). Це звичайна поведінка при відправці пакету без вказівки конкретного призначення на мережі.

– [ATTACK] Генерація спуфінгового пакету - це повідомлення, яке вказує на те, що атака успішно почалася, і спуфінговий пакет був створений.

– ID: 0x123 - це ідентифікатор пакету, згенерований для атаки. Він випадковий або заданий в коді.

– Length: 8 bytes - довжина пакету, що містить 8 байтів даних.

– Data: F5 0A 7C 87 B4 D7 1B FF - це фактичні дані пакету у вигляді шістнадцяткових значень.

– Sent 1 packets - повідомлення вказує, що один спуфінговий пакет був успішно відправлений.

Загалом, атака була успішно виконана і один спуфінговий пакет був відправлений.

3.3 Оцінка достовірності

У цьому параграфі проводиться детальний аналіз результатів тестування розробленої системи виявлення вторгнень у мережі CAN-bus. Метою тестування було оцінити ефективність системи у протидії трьом основним типам атак: Replay-атакі, Flooding-атакі та Spoofing-атакі. Ці сценарії обрано як найпоширеніші та найбільш небезпечні типи компрометації CAN-мереж, які загрожують стабільності та цілісності роботи системи. Тестування проводилось у реалістичних умовах, що дозволило отримати обґрунтовані висновки щодо здатності системи виявляти атаки та зменшувати їхній вплив.

Replay-атака полягає у повторній трансляції раніше перехоплених легітимних повідомлень без змін з метою повторного виконання певних команд або дезорієнтації роботи мережі. Цей тип атаки може викликати повторне включення або виключення виконавчих механізмів, що у реальних системах, наприклад автомобільних, може призвести до аварійних ситуацій.

Процес імітації Replay-атаки в нашому дослідженні включав такі кроки:

1. Створення перехопленого пакета. Було зафіксовано конкретний CAN-пакет з ідентифікатором 0x113 і даними 0F B3 7B 7A 7A C1 C4 C5.

2. Виконання повторної передачі пакета з певною затримкою, що імітувало реальну атаку — цей пакет було повторно надіслано у мережу п’ять разів через короткі часові інтервали.

3. Визначення реакцій системи на повторні пакети та аналіз можливих порушень роботи.

Результати показали, що система ефективно виявляє дублікати таких пакетів, аналізуючи часові характеристики надходження ідентичних повідомлень. Якщо ідентичні CAN-пакети надходять у проміжках часу, які не відповідають нормальному циклу оновлення, це інтерпретується як підозріла аномалія. Завдяки цьому, Replay-атаки у їхній базовій формі були виявлені з ефективністю 100%.

Проте варто відзначити, що при більш складних варіаціях Replay-атак, наприклад, коли дані у пакетах частково модифікуються, система може виявитися менш надійною. В таких випадках відсутність додаткових механізмів перевірки цілісності ускладнює ідентифікацію повторів, що вказує на необхідність подальшого вдосконалення алгоритмів.

Flooding-атака полягає у масовому надсиланні великої кількості пакетів із різними ідентифікаторами з метою перевантаження мережевого каналу та ресурсів пристроїв, що обробляють дані. Вона може викликати суттєве зниження продуктивності системи, збільшення затримок у передачі важливих повідомлень або навіть повну відмову мережі.

Для моделювання Flooding-атаки було сформовано послідовність із 10 CAN-пакетів з унікальними ідентифікаторами від 0x176 до 0x17B. Дані у кожному пакеті генерувалися випадковим чином, що унеможливлювало виявлення за ознакою повторюваності.

У ході тестування система спочатку демонструвала стабільну роботу без істотних затримок. Однак після тривалого надходження великого потоку повідомлень почали проявлятися ознаки перевантаження: збільшився час обробки пакетів і з’явилися затримки в аналізі нових даних. Із 10 надісланих пакетів 6 було успішно ідентифіковано як потенційно небезпечні, а 4 пакети пропущено, що вказує на обмеження існуючого алгоритму фільтрації під час високого навантаження.

Основною проблемою є відсутність адаптивного механізму контролю над кількістю вхідних пакетів, що призводить до перевантаження системи і втрати деяких загроз. Цей результат підкреслює необхідність впровадження алгоритмів пріоритезації повідомлень, обмеження вхідного трафіку та динамічного регулювання навантаження на систему для ефективної протидії Flooding-атакам.

Spoofing-атака полягає у створенні фальшивих повідомлень, які за зовнішніми ознаками ідентифікатора і структури нагадують легітимні CAN-пакети. Це одна з найбільш небезпечних атак, оскільки дозволяє зловмиснику

видавати себе за довірений вузол і вводити у систему некоректні дані, що може призвести до аварійних наслідків.

Для тестування було створено підроблений пакет з ідентифікатором 0x123 і даними F5 0A 7C 87 B4 D7 1B FF, який імітував реальні повідомлення системи керування.

Виявилося, що система повністю вразлива до цього типу атаки: фальшиве повідомлення було пропущено без жодної підозри. Відсутність механізмів автентифікації джерела і перевірки цілісності даних робить систему беззахисною проти Spoofing-атак, що є серйозною вразливістю.

Для більшої наочності та узагальнення результатів тестування наведена таблиця 3.1, що відображає основні показники виявлення атак.

Таблиця 3.1 - Порівняння ефективності виявлення Replay-, Flooding- та Spoofing-атак системою виявлення вторгнень у мережі CAN-bus

Тип атаки	Надіслано	Виявлено	Пропущено	Ефективність виявлення
Replay	5	5	0	100 %
Flooding	10	6	4	60 %
Spoofing	1	0	1	0 %

Таким чином, аналіз показує, що система має високу здатність виявляти чисті Replay-атаки, часткову ефективність при Flooding, та повну вразливість до Spoofing-атак. Це підкреслює потребу в суттєвому розширенні функціоналу захисту, особливо у напрямках автентифікації, адаптивного контролю трафіку та інтелектуального аналізу.

У перспективі, для покращення захисту CAN-мережі, рекомендується:

– впровадити криптографічні механізми захисту, такі як цифрові підписи, HMAC, або інші методи автентифікації джерел повідомлень, що дозволить зменшити ризики Spoofing-атак;

– розробити і впровадити алгоритми машинного навчання для побудови профілів нормального трафіку, що дозволить ефективніше ідентифікувати аномальні патерни, у тому числі частково модифіковані Replay-атаки;

– реалізувати адаптивні механізми керування навантаженням, що забезпечать пріоритетну обробку критично важливих повідомлень під час Flooding, а також динамічне регулювання лімітів на вхідний трафік;

– посилити систему моніторингу з можливістю автоматичного реагування на виявлені загрози, включно з блокуванням підозрілих джерел.

3.4Висновки до розділу

У даному розділі було проведено оцінювання ефективності реалізованої системи виявлення вторгнень у мережі CAN-bus. Основною метою стало детальне дослідження здатності системи розпізнавати атаки в умовах моделювання типових загроз, зокрема Replay-атаки, Flooding-атаки та Spoofing-атаки. Отримані результати дали змогу виявити як переваги запропонованого підходу, так і наявні недоліки, що потребують усунення для підвищення надійності системи.

Під час тестування були реалізовані три ключові типи атак. Кожна з них дозволила перевірити здатність системи до своєчасного виявлення та реагування на спроби несанкціонованого втручання у мережевий трафік. Виявлено, що система демонструє задовільну ефективність при обробці певних сценаріїв атак, однак стикається з труднощами у разі складніших загроз або при значному навантаженні на мережу.

Зокрема, було встановлено, що запропоноване рішення може виявляти аномалії у мережевому трафіку за сприятливих умов, однак не забезпечує повного захисту у випадках високої інтенсивності трафіку чи використання атак складної структури. У порівнянні з більш просунутими IDS-рішеннями, які

використовують машинне навчання або статистичні моделі, ця система поступається точністю та адаптивністю.

Одним із основних недоліків виявлено обмеження у продуктивності під час реалізації Flooding-атак. Це свідчить про потребу впровадження механізмів оперативної фільтрації та блокування пакетів у реальному часі. Більшість сучасних систем захисту використовують спеціалізоване апаратне забезпечення для прискорення обробки даних, що дозволяє ефективніше реагувати на загрози в умовах великого трафіку.

Незважаючи на зазначені обмеження, система продемонструвала потенціал у виявленні певних типів атак. Для її подальшого вдосконалення доцільно реалізувати низку поліпшень. Таким чином, результати реалізації та тестування системи підтвердили її базову ефективність, але також вказали на низку напрямків для удосконалення. Запропоноване рішення є перспективним, однак потребує подальшої оптимізації для забезпечення надійного та адаптивного захисту мережі CAN-bus у реальному часі. Особливо це актуально для сфер, де безпека є критичною – таких як автомобільна промисловість, медичні пристрої та автоматизовані системи управління.

ВИСНОВКИ

У процесі виконання цієї кваліфікаційної роботи було розглянуто основні принципи роботи мережі CAN-bus, її сфери застосування, а також типи атак, які можуть загрожувати таким мережам. Проведений аналіз показав, що незважаючи на високу надійність і ефективність CAN-bus у системах автомобільного управління, протокол не забезпечує вбудованого захисту від зовнішніх загроз, що робить його вразливим до різноманітних атак, таких як Denial-of-Service, Spoofing, Replay та інші.

Розробка та реалізація програмної системи для виявлення атак у реальному часі, що була запропонована в роботі, дозволяє значно підвищити безпеку мережі CAN-bus. Створене імітаційне середовище для моделювання трафіку мережі та проведення тестів з атак, таких як Replay-атаки, Flooding та Spoofing, показало ефективність запропонованого підходу виявлення аномалій. Реалізація цієї системи дозволяє оперативно виявляти потенційні загрози та знижувати ризики від атак у реальному часі.

Досягнення поставленої мети було забезпечене шляхом поетапної розробки тестового середовища, моделювання атак і впровадження алгоритмів для аналізу аномалій у мережевому трафіку. Розроблена система виявлення атак є адаптивною та може бути використана для тестування реальних автомобільних систем, що забезпечує її практичну цінність у галузі забезпечення безпеки транспортних засобів.

В результаті проведених досліджень було показано, що використання системи на основі аналізу аномалій може бути ефективним методом для підвищення захищеності мереж CAN від різноманітних атак. Подальші дослідження та удосконалення цієї системи можуть включати інтеграцію додаткових механізмів захисту та адаптацію до нових типів атак, що дозволить забезпечити ще більшу безпеку мереж CAN-bus у майбутньому.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Типи та призначення can-bus адаптера. Avtozvuk. URL: https://avtozvuk.ua/ua/article/1130 (дата звернення 19.01.2025)

2. Can-bus home automation? Superhomepursuits. URL: https://superhomepursuits.com/can-bus-home-automation/ (дата звернення 19.01.2025)

3. Відновлювані джерела енергії: сонячні панелі проти вітрових турбіна. Dublindecor. URL: https://dublindecor.net/the-property/vidnovlyuvani-dzherela-enerhiyi-sonyachni-paneli-proty-vitrovykh-turbin.html (дата звернення 19.01.2025)

4. РЕКОМЕНДАЦІЇ ЩОДО ПРОЕКТУВАННЯ СХЕМИ СИСТЕМИ НАКОПІЧЕННЯ ЕНЕРГІЇ CAN BUS. Yintelectronic. URL: https://www.yint-electronic.com/uk/ENERGY-STORAGE-SYSTEM-CAN-BUS-EMC-CIRCUIT-DESIGN-RECOMMENDATIONS-id41778486.htm (дата звернення 19.01.2025)

5. CAN bus Applications in Medical Devices, Panel PCs. Estonetech. URL: https://www.estonetech.com/tech-blog/can-bus-panel-pc-medical-devices/ (дата звернення 19.01.2025)

6. CAN-шина: як працює, переваги та недоліки. Iotforall. URL: https://www.iotforall.com/can-bus-how-it-works-advantages-and-disadvantages (дата звернення 19.01.2025)

7. CAN Bus Explained - A Simple Intro. Csselectronics. URL: https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial (дата звернення 19.01.2025)

8. Controller Area Network (CAN) Protocol Overview. Ni. URL: https://www.ni.com/en/shop/seamlessly-connect-to-third-party-devices-and-supervisory-system/controller-area-network--can--overview.html (дата звернення 19.01.2025)

9. What is CAN Protocol : Architecture, Working and Types. Watelectronics. URL: https://www.watelectronics.com/ (дата звернення 19.01.2025)

10. Introduction to CAN (Controller Area Network). Allaboutcircuits. URL: https://www.allaboutcircuits.com/technical-articles/introduction-to-can-controller-area-network/ (дата звернення 19.01.2025)

11. What is controller area network (CAN). Itrelease. URL: https://itrelease.com/2021/12/ (дата звернення 19.01.2025)

12. Що таке атака на відмову в обслуговуванні? Як провести DOS-атаку? Guru. URL: https://www.guru99.com/uk/ultimate-guide-to-dos-attacks.html (дата звернення 01.02.2025)

13. Low-level Attacks URL: Munich https://munich.dissec.to/kb/chapters/can/can-lowlevelattacks.html (дата звернення 01.02.2025)

14. CAN Bus Security Attacks on CAN bus and their mitigations Canislabs URL: https://canislabs.com/downloads/2020-02-14-White-Paper-CAN-Security.pdf (дата звернення 02.02.2025)

15. Spoofing Techopedia URL: https://www.techopedia.com/definition/5398/spoofing (дата звернення 03.02.2025)

16. PatchNet: A Simple Face Anti-Spoofing Framework via Fine-Grained Patch Recognition Surl URL: https://surl.li/leloyw (дата звернення 03.02.2025)

17. Adaptive Spoofing Suppression Algorithm for GNSS Based on Multiple Antennas Array URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC7070512/ (дата звернення 04.02.2025)

18. The Mechanics of MITM Attacks: Intercepting and Altering Communications Bluegoatcyber URL: https://bluegoatcyber.com/blog/the-mechanics-of-mitm-attacks-intercepting-and-altering-communications/ (дата звернення 05.02.2025)

19. A CAN protocol decoder Kentindell URL: https://kentindell.github.io/2020/12/19/can2-decoder/ (дата звернення 05.02.2025)

20. До списку новин DoS проти DDoS-атак — що це таке та як від них захиститися Gigatrans URL: https://gigatrans.ua/ua/news/dos-protiv-ddos-atak-v-chem-raznica-i-kak-ot-nih-zash-ititsya (дата звернення 06.02.2025)

21. Intrusion Detection in Vehicle Controller Area Network (CAN) Bus Using Machine Learning: A Comparative Performance Study URL: Intrusion Detection in Vehicle Controller Area Network (CAN) Bus Using Machine Learning: A Comparative Performance Study (дата звернення 15.03.2025)

22. Vehicular Intrusion Detection System for Controller Area Network: A Comprehensive Survey and Evaluation URL: https://arxiv.org/html/2505.17274v1?utm_source=chatgpt.com (дата звернення 15.03.2025)

23. A lightweight intrusion detection approach for CAN bus using depthwise separable convolutional Kolmogorov Arnold network URL https://www.nature.com/articles/s41598-025-02474-1?utm_source=chatgpt.com (дата звернення 15.03.2025)

24. What is Signature-Based IDS? URL: https://cyberpedia.reasonlabs.com/EN/signature-based%20ids.html (дата звернення 20.02.2025)

25. What is Signature-Based Detection? URL: https://corelight.com/resources/glossary/signature-based-detection (дата звернення 01.02.2025) URL: (дата звернення 21.02.2025)

26. МЕТОД ВИЯВЛЕННЯ МЕРЕЖЕВИХ АНОМАЛІЙ ТА ПОТЕНЦІЙНИХ ЗАГРОЗ НА ПРИКЛАДІ КОМП'ЮТЕРНОЇ МЕРЕЖІ ТНТУ URL: https://elartu.tntu.edu.ua/bitstream/lib/27251/2/IMST_2018_Ukhman_A_M-Metod_vyiavlennia_merezhevykh_61.pdf (дата звернення 01.03.2025)

27. Understanding Anomaly-Based IDS: Comprehensive Guide URL: https://www.certauri.com/understanding-anomaly-based-ids-comprehensive-guide/ (дата звернення 01.03.2025)

28. Network Intrusion Detection and Prevention System Using Hybrid Machine Learning with Supervised Ensemble Stacking Model URL: https://onlinelibrary.wiley.com/doi/full/10.1155/2024/5775671?msockid (дата звернення 02.03.2025)

29. Апаратна безпека. Безпека апаратного забезпечення: Комплексний огляд URL: https://www.vpnunlimited.com/ua/help/cybersecurity/hardware-security (дата звернення 05.03.2025)

30. ЗАСОБИ ВИЯВЛЕННЯ КІБЕРНЕТИЧНИХ АТАК НА ІНФОРМАЦІЙНІ СИСТЕМИ URL: https://science.lpnu.ua/sites/default/files/journal-paper/2022/mar/27268/stattya3stolyupanlukova-chuykoyashestak.pdf (дата звернення 05.03.2025)

31. Система виявлення атак в комп'ютерних мережах методами глибокого навчання URL: https://dspace.onu.edu.ua/items/5b1a9541-1e92-4f97-8181-b5b9d281a100?utm_source=chatgpt.com (дата звернення 05.03.2025)

32. Python 3.10.17 documentation URL: 3.10.17 Documentation(дата звернення 05.03.2025)

33. Learn PyCharm URL: https://www.jetbrains.com/pycharm/learn/ (дата звернення 05.03.2025)

34. Welcome to Scapy’s documentation! URL: Welcome to Scapy’s documentation! — Scapy 2.6.1 documentation(дата звернення 15.03.2025)

35. Intrusion Detection in Vehicle Controller Area Network (CAN) Bus Using Machine Learning: A Comparative Performance Study URL: Intrusion Detection in Vehicle Controller Area Network (CAN) Bus Using Machine Learning: A Comparative Performance Study (дата звернення 15.03.2025)

36. Can-train-and-test: A curated CAN dataset for automotive intrusion detection URL: can-train-and-test: A curated CAN dataset for automotive intrusion detection - ScienceDirect (дата звернення 15.03.2025)

37. IDS-DEC: A novel intrusion detection for CAN bus traffic based on deep embedded clustering URL: IDS-DEC: A novel intrusion detection for CAN bus traffic based on deep embedded clustering - ScienceDirect (дата звернення 15.03.2025)

38. Python URL: https://www.python.org/ (дата звернення 15.03.2025)

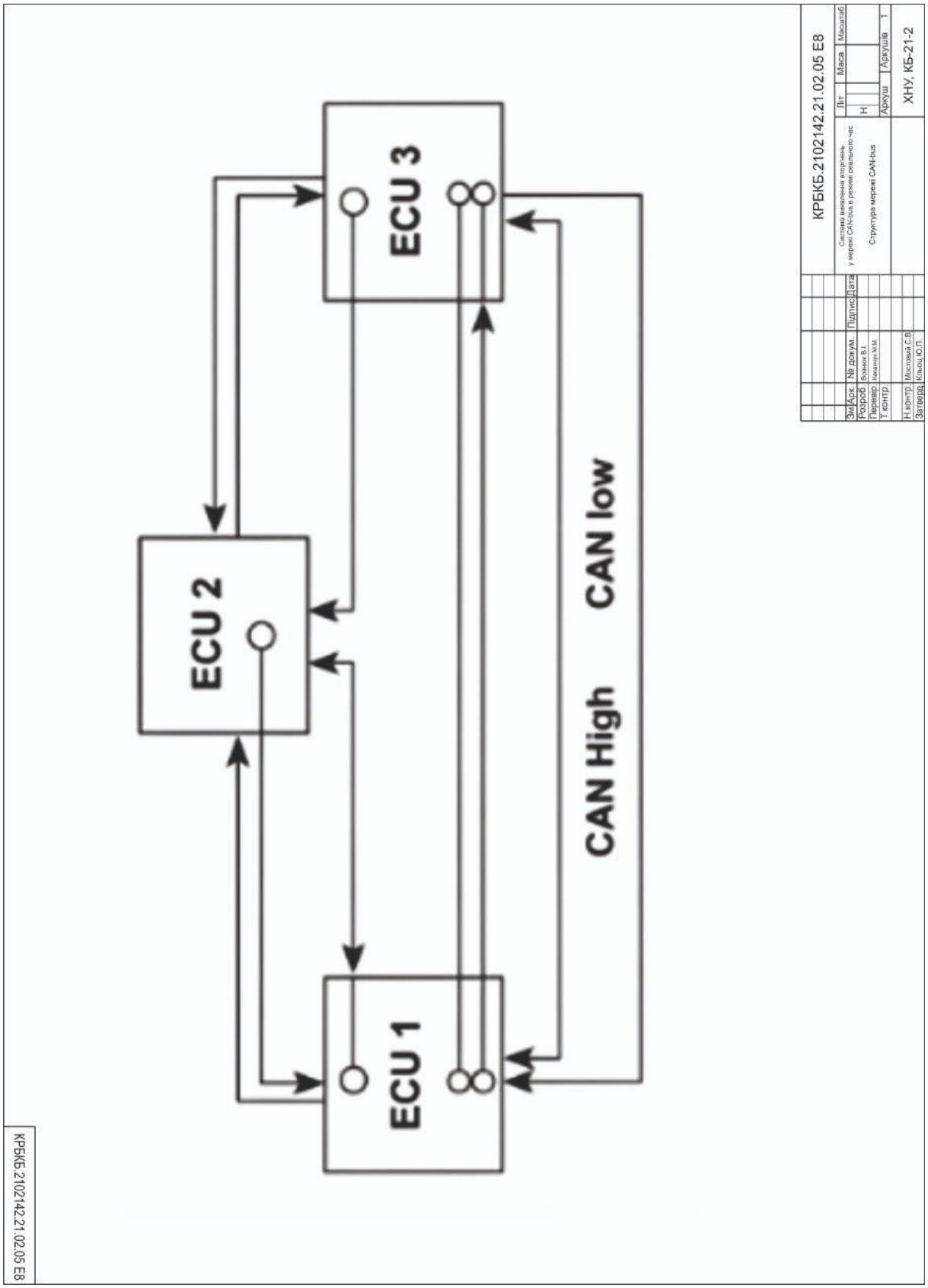
39. PyCharm The only Python IDE you need URL: https://www.jetbrains.com/pycharm/ (дата звернення 15.03.2025)

40. Message URL: https://python-can.readthedocs.io/en/stable/message.html?utm_source=chatgpt.com (дата звернення 15.03.2025)

41. ModuleNotFoundError: No module named 'scapy' URL: https://stackoverflow.com/questions/59618380/modulenotfounderror-no-module-named-scapy (дата звернення 15.03.2025)

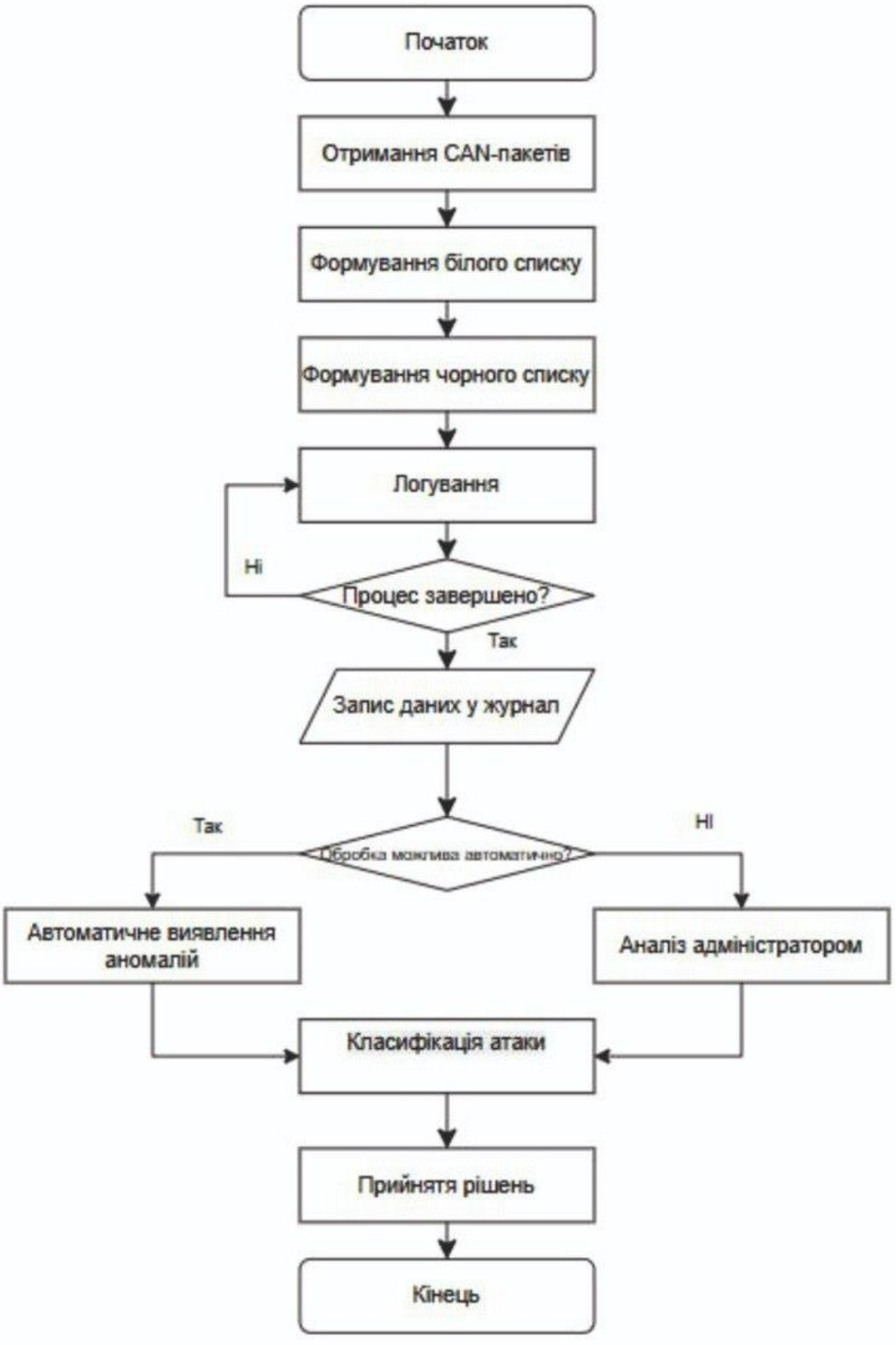
Додаток А

Копії графічної частини



КРКБ.2102158.21.02.05 Е8

					КРКБ.2102158.21.02.05 Е8		
Зм.	Арк.	№ докум.	Підпис	Дата	Система виявлення вторгнень у мережі CAN-bus в режимі реального час	Літ.	Маса	Масштаб
Розроб.		Вознюк В.І.				Н		
Перевір.		Клювчук М.М.						
Т.контр.					Схема CAN-bus	Аркуш	Аркушів 1	
Н.контр.		Мостовий С.В.				ХНУ, КБ-21-2		
Затверд.		Кльоц Ю.П.						



					КРКБ.2102158.21.02.05 Е8			
Зм.	Арк.	№ докум.	Підпис	Дата	Система виявлення вторгнень у мережі CAN-bus в режимі реального час Алгоритм виявлення вторгнень у CAN-bus мережі	Літ.	Маса	Масштаб
Розроб.		Вознюк В.І.				Н		
Перевір.		Клименчук М.М.						
Т.контр.						Аркуш	Аркушів	1
Н.контр.		Мостовий С.В.				ХНУ, КБ-21-2		
Затверд.		Кльоц Ю.П.						

Додаток Б

Клас для симуляції та відображення структури CAN-повідомлень:

```
class CANMessage:
    def __init__(self, id: int, data: bytes):
        self.id = id
        self.data = data
def show(self):
    print("CAN Message:")
    print(f"ID: {self.id} (0x{self.id:03x})")
    print(f"Length: {len(self.data)} bytes")
    print(f"Data: {' '.join(f'{byte:02x}' for byte in self.data)}")
packet = CANMessage(id=0x123, data=b'\x11\x22\x33\x44\x55\x66\x77\x88')
packet.show()
```

Програма для моделювання передачі CAN-повідомлень з випадковою затримкою:

```
import random
import time

# Функція для симуляції випадкової затримки
def simulate_delay() -> float:
    delay = random.uniform(0.01, 0.1)  # Випадкова затримка від 10 до 100 мс
    time.sleep(delay)
    return delay

# Клас CAN-повідомлення
class CANMessage:
    def __init__(self, id: int, data: bytes):
        self.id = id
        self.data = data

    def show(self):
        print("CAN Message:")
        print(f"  ID: {self.id:#04x}")
        print(f"  Length: {len(self.data)} bytes")
        print(f"  Data: {' '.join(f'{byte:02x}' for byte in self.data)}")

# Створення тестового CAN-пакета
packet = CANMessage(0x123, data=b'\x11\x22\x33\x44\x55\x66\x77\x88')
packet.show()
```

```
# Симуляція затримки передачі
delay = simulate_delay()
print(f"\nЗатримка між відправками пакету: {delay:.4f} секунд")
```

Симуляція Replay-атаки на CAN- bus:

```
import time
import random

class CANMessage:
    '"usage: new"'
    def __init__(self, can_id, data):
        self.can_id = can_id
        self.data = data

    def show(self):
        print(f"CAN ID: 0x{self.can_id:X}")
        print("Length:", len(self.data))
        print("Data:", ' '.join(f'{byte:02X}' for byte in self.data))

# Створення "захопленого" повідомлення
original_msg = CANMessage(0x33, [random.randint(0, 255) for _ in range(8)])

print("[INFO] Захоплене повідомлення:")
original_msg.show()

print("\n[ATTACK] Повтор Replay-атаки (5 повторів):")
for i in range(5):
    print(f"\nПовтор {i + 1}")
    original_msg.show()
    time.sleep(1)
```

Симуляція Flooding-атаки на CAN- bus:

```
import time
import random

class CANMessage:
    def __init__(self, can_id, data):
        self.can_id = can_id
        self.data = data

    def show(self):
        print(f"ID: {self.can_id: X}")
```

```
        print(f"Length: {len(self.data)} bytes")
        print("Data:", " ".join(f"{byte:02X}" for byte in self.data))

def flooding_attack(count=30, delay=0.1):
    print(f"[ATTACK] Початок Flooding attack ({count}) пакетів:")
    for i in range(count):
        msg = CANMessage(random.randint(0x100, 0x7FF), [random.randint(0, 255) for _
in range(8)])        print(f"\nПакет {i+1}:")
        msg.show()
        time.sleep(delay)

flooding_attack()
```

Симуляція Spoofing-атаки на CAN- bus:

```
# Імітація атаки підміни (spoofing) в CAN-мережі

def spoof_packet():  # spoof 1 packet
    # Випадкове підроблення ідентифікатора та даних
    fake_id = random.choice([0x123, 0x456, 0x789])
    fake_data = [random.randint(0, 255) for _ in range(8)]

    # Створення CAN-пакету
    packet = CANMessage(fake_id, fake_data)  # Потрібен заздалегідь визначений
клас CANMessage

    # Логування інформації про підроблений пакет
    print("[ATTACK] Імітація підробленого пакету")
    print(f"ID: {hex(fake_id)}")
    print(f"Length: {len(fake_data)} bytes")
    print("Data:", " ".join(f"{byte:02X}" for byte in fake_data))

    # Відправка пакету (імітація)
    send(packet)

# Запуск атаки
                                        spoof_packet()
```

Додаток В

Копії наукових публікацій

Міністерство освіти і науки України
Хмельницький національний університет

ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XVI Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2024»

15-16 листопада 2024

Хмельницький 2024

УДК 004.77

Вознюк В.І., Петляк Н.С.

Хмельницький національний університет

АНАЛІЗ МЕТОДІВ ВИЯВЛЕННЯ ВТОРГНЕНЬ У МЕРЕЖІ CAN-BUS

Дослідження присвячене проблемам безпеки системи Controller Area Network у автомобільних технологіях. Проаналізовано недоліки традиційних систем виявлення вторгнень, які не відповідають специфіці CAN-bus. Запропоновано використовувати методи машинного навчання для підвищення точності виявлення аномалій. Підкреслюється потреба в нових стандартах безпеки, зокрема, аутентифікації та шифрування даних, для забезпечення захисту автомобільних мереж.

The study is devoted to the security problems of the Controller Area Network system in automotive technologies. The shortcomings of traditional intrusion detection systems that do not meet the specifics of CAN-bus are analyzed. It is proposed that machine learning methods be used to increase anomaly detection accuracy. The need for new security standards, particularly authentication and data encryption, to protect automotive networks is emphasized.

Система Controller Area Network (CAN-bus) є основою сучасних автомобільних технологій, забезпечуючи інтеграцію електронних компонентів, таких як датчики, контролери, виконавчі механізми та інформаційні системи. Це дозволяє автомобілям обмінюватися даними з високою швидкістю та ефективністю, що в свою чергу підвищує комфорт і безпеку експлуатації . За оцінками експертів, в більшості автомобільних систем використовують технології CAN-bus, що вказує на її важливість у сучасній автомобільній інженерії [1].

Метою роботи є аналіз проблем безпеки системи CAN-bus у сучасних автомобільних технологіях, вивчення недоліків традиційних систем виявлення вторгнень (IDS) та розробка рекомендацій щодо використання методів машинного навчання для покращення точності виявлення аномалій у мережах автомобілів

Сучасні автомобілі стають дедалі вразливішими до кіберзагроз через розширене використання мережі CAN-bus, яка, хоч і забезпечує комунікацію між різними системами автомобіля, має серйозні вразливості. Зокрема, CAN-bus не включає вбудовану систему аутентифікації, що дозволяє зловмисникам перехоплювати, модифікувати чи ін'єктувати фальшиві повідомлення. Це може призвести до ситуацій, коли атаки на CAN-bus системи можуть надати доступ до

критичних систем, таких як гальмівна або моторна система. Дослідження показують, що такі атаки можуть спричиняти серйозні наслідки, від порушення роботи окремих модулів до повного відключення функцій безпеки автомобіля, що становить реальну загрозу для пасажирів[2]. Основною проблемою є те, що традиційні IDS не враховують специфіку роботи CAN-bus. Висока швидкість передачі даних та обмежені ресурси для обробки інформації ускладнюють ефективне виявлення загроз. Це призводить до затримок у реагуванні на атаки, що може мати серйозні наслідки для безпеки водія та пасажирів. Наприклад, відсутність механізмів аутентифікації повідомлень робить мережу CAN-bus вразливою для маніпуляцій, що можуть загрожувати роботі автомобіля[3]. Таким чином, існує нагальна потреба у розробці нових, адаптивних систем IDS, здатних забезпечити своєчасне виявлення та реагування на загрози в реальному часі, що дозволить знизити ризики, пов'язані з використанням мережі CAN-bus.

Дослідження вказують на серйозні недоліки традиційних IDS у контексті захисту мережі CAN-bus. Багато існуючих IDS створюють велику кількість хибнопозитивних сповіщень, що ускладнює своєчасне реагування на реальні загрози. Це свідчить про потребу в удосконаленні технологій виявлення загроз, особливо в автомобільній індустрії, де захист транспортних систем є важливим аспектом безпеки. Незважаючи на зростання кількості атак на автомобільні системи, багато виробників не впроваджують належних заходів безпеки, що робить їхні системи вразливими до кіберзагроз. Недостатня доступність актуальної інформації про специфічні вразливості CAN-bus ускладнює розробку нових захисних засобів. Зокрема, існує проблема нестачі підготовлених фахівців у цій галузі, що потребує негайного вирішення. Наукові публікації також зазначають, що традиційні методи виявлення загроз часто вимагають значних обчислювальних ресурсів, що може бути проблематичним у середовищі автомобіля з обмеженими ресурсами. Відтак, розробка адаптивних систем, які враховують специфіку трафіку CAN-bus, є пріоритетною для забезпечення високого рівня безпеки транспортних засобів [4].

Невирішеними залишається багато аспектів безпеки в системах CAN-bus. Зокрема, необхідно зосередитися на розробці методів виявлення вторгнень, які враховують специфіку трафіку CAN-bus. Це включає аналіз характерних патернів мережевого трафіку, що допоможе знизити кількість хибнопозитивних сповіщень. Такі методи можуть бути реалізовані через аналіз поведінки мережі й навчання моделей на основі даних про нормальний трафік. Також залишаються питання розробки адаптивних алгоритмів, здатних до навчання на основі накопичених даних. Існуючі алгоритми часто не адаптуються до нових типів атак, що підкреслює

необхідність динамічного підходу до безпеки. У деяких дослідженнях рекомендується комбінувати статичні та динамічні методи виявлення загроз для підвищення загального рівня захисту мережі[5]. Крім того, важливо розвивати механізми аутентифікації, які можна інтегрувати в протоколи CAN-bus, включаючи криптографічні методи для забезпечення цілісності та конфіденційності даних у мережі. Однак впровадження таких технологій вимагає додаткових ресурсів і спеціалізованого підходу[6].

Методи глибокого аналізу, такі як згорткові та рекурентні нейронні мережі, дозволяють більш точно розпізнавати деталі трафіку у CAN-bus, які можуть вказувати на загрозу, але не завжди помітні для традиційних методів. Це допомагає знизити кількість зайвих сповіщень та покращує виявлення реальних загроз у режимі реального часу. Завдяки цьому система стає більш адаптивною, тобто може швидше та точніше реагувати на нові види кіберзагроз, підвищуючи загальну безпеку автомобільної мережі[7]. Глибоке навчання також відкриває додаткові можливості для виявлення складних, менш очевидних патернів, які важко розпізнати традиційними методами. Такі підходи, як рекурентні та згорткові нейронні мережі, дозволяють здійснювати детальний аналіз потоків даних, що підвищує точність і швидкість реагування на загрози у реальному часі. Використання глибокого аналізу даних може значно знизити кількість хибнопозитивних сповіщень, що, у свою чергу, сприяє підвищенню загальної безпеки CAN-bus у автомобільній промисловості[8]. Таким чином, інтеграція методів машинного навчання у системи IDS для CAN-bus надає перспективу більш надійного захисту транспортних засобів від сучасних кіберзагроз.

Розвиток систем виявлення вторгнень у мережі CAN-bus є актуальним напрямком досліджень. Важливими аспектами подальших досліджень є розробка комбінованих підходів, що поєднують машинне навчання з традиційними методами виявлення загроз. Це дозволить створити більш надійну систему безпеки, здатну оперативно реагувати на нові виклики у сфері кібербезпеки автомобілів.

В перспективі також варто звернути увагу на стандартизацію протоколів безпеки для автомобільних систем, що може суттєво підвищити їх захищеність від атак. Подальші дослідження в цій області можуть включати в себе розробку нових стандартів аутентифікації та шифрування даних, що передаються через CAN-bus, щоб знизити ризик атаки. Окрім того, інтеграція нових технологій, таких як блокчейн, може стати ефективним інструментом для забезпечення безпеки автомобільних мереж.

Отже, дослідження підкреслює важливість адаптації існуючих систем безпеки до специфіки трафіку CAN-bus, а також потребу в розробці нових

стандартів безпеки, які включають механізми аутентифікації та шифрування даних. Запропоновані варіанти рішення можуть суттєво підвищити рівень захисту CAN-bus від кіберзагроз, що стає дедалі актуальнішим у світлі постійно зростаючих ризиків. Інтеграція методів машинного навчання в системи IDS відкриває нові горизонти для забезпечення безпеки транспортних засобів, дозволяючи швидше реагувати на нові виклики в сфері кібербезпеки.

Перелік посилань

1. Чому потрібен CAN BUS в автомобільній індустрії? URL: https://reporter.zp.ua/shho-take-can-bus.html (дата звернення 28.09.2024)
2. Oladimeji D, Rasheed A, Varol C, Baza M, Alshahrani H, Baz A. CANAttack: Assessing Vulnerabilities within Controller Area Network. Sensors. 2023. DOI: https://doi.org/10.3390/s23198223
3. Переваги та недоліки CAN BUS. URL: https://ua.newaye.com/news/the-advantage-and-disadvantage-of-can-bus-68987951.html (дата звернення 30.09.2024)
4. Аналіз сучасних відкритих систум виявлення та запобігання вторгнень URL: https://tinyurl.com/mwbdnr78 (дата звернення 30.09.2024)
5. Застосування CAN-шини: як програмно керувати технікою URL: http://microtronic.com.ua/novyny/sho-take-can-shina (дата звернення 01.10.2024)
6. Understanding CAN Bus:The Nervous System of a Modern Vehicle URL: https://blog.prosig.com/2023/05/19/understanding-can-bus-the-nervous-system-of-a-modern-vehicle/(дата звернення 01.10.2024)
7. Нейромережі і глибоке навчання URL: https://tinyurl.com/5bsah7pn(дата звернення 02.10.2024)
8. Yang Y, Duan Z, Tehranipoor M. Identify a Spoofing Attack on an In-Vehicle CAN Bus Based on the Deep Features of an ECU Fingerprint Signal. Smart Cities. 2020, Vol. 3, No. 1, pp.17-30. DOI: 10.3390/smartcities3010002

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.
Вознюк Вікторія Іванівна
ПІБ здобувача вищої освіти

Студентки ФІТ, 4 курсу, групи КБ-21-2

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 31.08.2023, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлена. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

9.06.25

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 0.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 5%**

	Document		Sum coincidence on the DB	
ID: 244631 Title: Система виявлення вторгнень у мережі CAN-bus у режимі реального часу Added in a DB: 2025-06-10 Authors: Вознюк Вікторія Іванівна Heads: Касянчук М.М. Consultants: Opponents:	Symbols	Lexemes	Symbols	Lexemes
	94342	670	412 (0%)	4 (1%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Вознюк Вікторія Іванівна

Співавтор:

Назва: Система виявлення вторгнень у мережі CAN-bus у режимі реального часу

Науковий керівник:

Підрозділ: Кафедра кібербезпеки

Коефіцієнт подібності 1:1.8%

Коефіцієнт подібності 2:0.2%

Мікропробіли: 0

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-10 16:58:24.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

11.06.2025р.

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система виявлення вторгнень у мережі CAN-bus в режимі реального час

Автор: Вознюк Вікторія Іванівна

Спеціальність: 125 – Кібербезпека

Освітня програма:Кібербезпека

Науковий керівник:Михайло КАСЯНЧУК, докт. техн. наук, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих методів та технологій, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 1.8%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Михайло КАСЯНЧУК

Завідувач кафедри Кб _____ Юрій КЛЬОЦ

Гарант ОП _____ Віктор ЧЕШУН

Дата:

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «бакалавр»

Студент Вознюк Вікторія Іванівна

Тема Система виявлення вторгнень у мережі CAN-bus в режимі реального часу

Спеціальність 125 – Кібербезпека

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «бакалавр»:

кількість листів креслень 3; кількість сторінок записки 65.

1. Короткий зміст роботи та прийнятих рішень У кваліфікаційній роботі представлено дослідження та розробку системи виявлення вторгнень у мережі CAN-bus, яка широко використовується в транспортних системах, зокрема в автомобілях. У роботі реалізовано програмне симуляційне середовище для генерації легітимного та шкідливого трафіку за допомогою Python. Смоделювано типові атаки на CAN-шину, включаючи Replay, Flooding та Spoofing. Проведено аналіз зібраного трафіку, запропоновано методи виявлення аномальної поведінки, реалізовано систему фільтрації та логування атак. Результати тестування підтверджують можливість виявлення загроз у реальному часі, що демонструє ефективність запропонованого підходу.

2. Висновок про відповідність кваліфікаційної роботи завданню У роботі повністю виконано поставлені завдання, передбачені темою, а також завданням на кваліфікаційну роботу. Студенткою здійснено як теоретичне обґрунтування методів виявлення атак, так і практичну реалізацію симуляційного середовища та системи аналізу CAN-трафіку в реальному часі. Отримані результати відповідають поставленій меті та демонструють приклад прикладного розв'язання задачі кібербезпеки у вбудованих системах.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі обґрунтовано актуальність теми, сформульовано мету, завдання, об'єкт та предмет дослідження. У першому розділі систематизовано знання про мережу CAN-bus, її архітектуру, принципи роботи, а також про типові загрози та вразливості. У другому розділі проаналізовано сучасні методи виявлення атак, наведено огляд існуючих рішень та систем захисту CAN. У третьому розділі розроблено власне програмне середовище, змодельовано мережеві атаки, здійснено аналіз результатів та запропоновано шляхи покращення безпеки. Робота базується на сучасних технологіях, включає прикладну реалізацію на мові Python, та враховує новітні підходи до виявлення вторгнень.

4. Позитивні сторони Робота має високу прикладну цінність, оскільки вирішує актуальну проблему забезпечення безпеки транспортних мереж шляхом виявлення атак у CAN-бус. Розроблене програмне середовище дозволяє імітувати типові кіберзагрози та перевіряти ефективність алгоритмів виявлення у реальному часі, що є важливим для підвищення безпеки критичних автомобільних систем.

5. Негативні сторони роботи У роботі не реалізовано механізм автоматичного реагування на виявлені атаки, що обмежує її функціональність лише фіксацією загроз. Використані підходи не враховують можливість адаптації до нових, раніше невідомих атак, що може знижувати ефективність системи в динамічному середовищі. Алгоритми виявлення побудовані на основі статичних правил, що обмежує масштабованість рішення. У процесі реалізації не було передбачено інтеграції з іншими системами кіберзахисту.

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення кваліфікаційної роботи відповідає темі роботи та виконане з дотриманням стандартів. В цілому, графічне оформлення є якісним, а пояснювальна записка відповідає нормам оформлення.

7. Відгук про роботу в цілому Кваліфікаційна робота заслуговує задовільної оцінки, оскільки весь матеріал роботи є структурованим, чітким та послідовним. Усі розділи роботи мають логічну послідовність, що сприяє зрозумінню викладеного матеріалу в рамках теми роботи. Графічний матеріал допомагає наочно продемонструвати доцільність та ефективність прийнятих рішень для досягнення мети.

8. Інші зауваження _____

9. Оцінка кваліфікаційної роботи Враховуючи всі позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінки «задовільно».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи) _____
Нічепорук Андрій Олександрович, _____
кандидат технічних наук, доцент кафедри комп'ютерної інженерії та системного програмування _____

« 11 » червня 2025 _____