КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ РАДІОФІЗИКИ ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ

Кафедра радіотехніки та радіоелектронних систем

До захисту допущено: «На правах рукопису»
Завідувач кафедри _____ Ігор АНІСІМОВ
18 травня 2023 р.

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

«**Виявлення атак на пристрої Bluetooth за допомогою машинного навчання**»

Виконала:
студентка 2-го курсу магістратури
денної форми навчання
спеціальності 172 Телекомунікації та радіотехніка
ОНП «Інформаційна безпека телекомунікаційних систем і мереж»
Долінчук Катерина Олександрівна _____

Науковий керівник:
канд. тех. н., доц. Жиров Геннадій Борисович _____

Рецензент:
д. т. н., проф. Вишнівський Віктор Вікторович _____

Засвідчую, що у цій магістерській роботі
немає запозичень з праць інших авторів без
відповідних посилань
Студент _____

Робота допущена до захисту в ЕК рішенням кафедри радіотехніки та радіоелектронних систем від 18 травня 2023 р., протокол № 18.

Завідувач кафедри радіотехніки та радіоелектронних систем,
доктор фіз.-мат. наук, професор
Анісімов Ігор Олексійович _____

Київ – 2023

ЗМІСТ

ВСТУП

Інтернет речей (IoT) - це мережа підключених до Інтернету пристроїв, яка нараховує мільярди звичайних побутових та промислових об'єктів. Зараз налічується понад 7 мільярдів підключених пристроїв IoT, але передбачається, що ця кількість зросте до 22 мільярдів до 2025 року. Незважаючи на такий стрімкий ріст, виробники часто не приділяють достатню увагу безпеці цих пристроїв, при цьому 98% трафіку IoT залишається незашифрованим. Особливо вразливими є пристрої, що використовують технологію Bluetooth, такі як фітнес-трекери, старт-годинники, Bluetooth-гарнітури та інші. Наприклад, деякі фітнес-трекери можуть залишатися видимими після першого сполучення, розумний холодильник може викладати облікові дані для входу в Gmail, а розумний замок із відбитками пальців можна розблокувати за допомогою ключа Bluetooth, який має ту саму MAC-адресу, що й пристрій замка. Це є однією з найбільших проблем безпеки IoT [1].

Наявність даної проблеми вимагає розробки системи виявлення несанкціонованих дій щодо пристроїв, що підтримують технологію Bluetooth.

Мета і завдання дослідження. Метою роботи є підвищення ефективності системи моніторингу атак на пристрої Bluetooth. Для досягнення поставленої мети необхідно розв’язати актуальне науково-технічне завдання, щодо розробки програми, яка здатна виявляти можливі атаки на пристрої Bluetooth з використанням методів машинного навчання.

Об’єкт дослідження : атаки на пристрої Bluetooth

Предмет дослідження: модель машинного навчання для виявлення атак на пристрої Bluetooth.

Завдання роботи:

1. Огляд та аналіз особливостей технології Bluetooth.
2. Вивчення недоліків в безпеці технології : огляд різних атак, вразливостей та вірусів .

3. Дослідження методів здійснення атак на пристрої Bluetooth за допомогою ОС Kali Linux.
4. Розробка моделі машинного навчання для виявлення атак на пристрої Bluetooth.
5. Впровадження готової моделі машинного навчання в програму для сканування пристроїв на виявлення атак.

Загальний обсяг роботи – 69 сторінок друкованого тексту. Основний текст викладений на 55 сторінках.

У розділі I проаналізовані основні поняття і визначення технології Bluetooth, особливості її побудови та роботи.

У розділі II проведений аналіз основних вразливостей технології Bluetooth, способів їх реалізації та ознак здійснення.

У розділі III наведений програмний код створеного додатку для виконання задачі виявлення атак на пристрої Bluetooth.

У висновках підсумовані основні результати дослідження.

ТЕХНОЛОГІЯ BLUETOOTH

1.1. Історія винайдення

Bluetooth – це бездротова технологія зв’язку, яка дозволяє підключати пристрої один до одного на відстані до кількох метрів за допомогою радіохвиль.

В 1994 році телекомунікаційна компанія ERICSSON розпочала дослідження можливостей створення життєдіяльного недорого радіо інтерфейсу між мобільними телефонами та аксесуарами. Головною метою була реалізація ідеї позбутися кабелів між мобільними телефонами та картками Пк, гарнітурами, настільними пристроями, тощо. Мотив був простий: для того, щоб система була успішною і дійсно придатною для використання, критична кількість портативних пристроїв повинна використовувати одну й ту саму технологію..

У лютому 1998 років було засновано групу особливих інтересів (SIG) компаніями Ericsson, Nokia, IBM, Toshiba та Intel. Ця група мала ідеальну комбінацію для бізнес-сфери: двох лідерів ринку мобільної телефонії, двох лідерів ринку ноутбуків та лідера ринку у технології обробки цифрових знаків. Метою було створення глобальної специфікації для бездротового зв’язку на короткій відстані. 20 та 21 травня 1998 року, консорціум Bluetooth був представлений громадськості в Лондоні (Англія), Сан-Хосе (Каліфорнія) та Токіо (Японія). Це глобальне представлення стимулювало інші компанії впроваджувати нову технологію для своїх продуктів. У 1999 році було прийнято рішення про створення єдиного стандарту бездротового зв’язку [2].

Коли стало питання про назву технології, Джим Кардач з компанії Intel запропонував назву технології Bluetooth в честь датського короля Харальда Блатанда, який об'єднав Данію, Норвегію та Швецію, і був відомий як "синій зуб" через гнилий синюватий зуб.

1.2. Характеристики Bluetooth

Основне завдання Bluetooth полягає в тому, щоб забезпечувати надійний та економний радіозв'язок між різними електронними пристроями, такими як мобільні телефони, портативні та настільні комп'ютери, принтери та інші, за допомогою компактних електронних компонентів, що дозволяє застосовувати Bluetooth у невеликих пристроях різного розміру, включаючи наручні годинники. Bluetooth дозволяє обмінюватись інформацією між пристроями, такими як персональні комп'ютери, мобільні телефони, ноутбуки, принтери, цифрові фотоапарати, мишки, клавіатури, джойстики, навушники, гарнітури на близькій радіочастоті за допомогою недорогого та надійного зв'язку. Залежно від механічних та радіо перешкод, дальність зв'язку може складати від 10 до 100 метрів, включаючи можливість передачі сигналів через стіни та інші перешкоди [3].

Технологія Bluetooth була створена з метою забезпечення недорогого, стійкого, ефективного та високопродуктивного зв'язку для передачі даних і голосу. Основні характеристики цієї технології включають наступне:

1. Швидкість передачі/прийому даних складає 1 Мбіт/с з використанням каналу з максимальною шириною смуги.
2. Швидке перемикання частот для уникнення інтерференції.
3. Адаптивна вихідна потужність для мінімізації перешкод.
4. Короткі пакети даних для економії енергії під час передачі даних.
5. Швидке підтвердження (впізнання).
6. Голосове кодування CVSD (Continuous Variable Slope Delta Modulation), що дозволяє працювати з високим рівнем помилок на біти.
7. Низькі витрати на зв’язок, що дозволяє додавати окремі елементи без значних витрат.

8. Інтерфейс передачі/прийому, спеціально розроблений для мінімізації споживання енергії [4].

Технологія Bluetooth може забезпечити надзвичайно гнучкий зв'язок з високою швидкістю передачі даних, навіть при наявності серйозних перешкод. При отриманні сильного сигналу в сприятливих умовах передачі, якість сигналу залишатиметься максимальною. Але прийом-передача буде відбуватися при наявності перешкод, падіння якості буде поступовим і мінімальним, що дозволяє зберегти стабільний зв'язок.

Таблиця 1.1 .

Порівняльна характеристика різних редакцій Bluetooth

Версія Bluetooth	Швидкість передачі даних	Дальність передачі	Сумісність	Споживання енергії	Підтримка нововведень
Bluetooth 1.0	1 Мбіт/с	10 метрів	Сумісний з Bluetooth 1.0	Високе споживання енергії	Немає
Bluetooth 2.0	3 Мбіт/с	10 метрів	Сумісний з Bluetooth 1.0 та Bluetooth 2.0	Високе споживання енергії	EDR
Bluetooth 2.1	3 Мбіт/с	10 метрів	Сумісний з Bluetooth 1.0, Bluetooth 2.0 і Bluetooth 2.1	Нижче споживання енергії	SSP, AVRCP
Bluetooth 3.0	24 Мбіт/с	10 метрів	Сумісний з Bluetooth 2.0 і Bluetooth 2.1	Нижче споживання енергії	HS, AMP
Bluetooth 4.0	25 Мбіт/с	50 метрів	Сумісний з Bluetooth 3.0 і Bluetooth 4.0	Дуже низьке споживання енергії	LE, Smart Ready
Bluetooth 4.1	25 Мбіт/с	100 метрів	Сумісний з Bluetooth 4.0 та Bluetooth 4.1	Дуже низьке споживання енергії	EDR, LE
Bluetooth	25 Мбіт/с	100 метрів	Сумісний з	Дуже низьке	Smart Ready,

4.2			Bluetooth 4.0, Bluetooth 4.1 і Bluetooth 4.2	споживання енергії	IPv6, LE
Bluetooth 5.0	2 Мбіт/с	200 метрів	Сумісний з Bluetooth 4.0, Bluetooth 4.1, Bluetooth 4.2 і Bluetooth 5.0	Дуже низьке споживання енергії	Smart Ready, LE, Mesh

Як видно з Таблиці 1.1, кожна нова версія Bluetooth відрізняється від попередньої швидкістю передачі даних, дальністю передачі, сумісністю з попередніми версіями, рівнем споживання енергії та профілями Bluetooth, які підтримуються.

1.3. Принцип дії Bluetooth

Радіозв'язок Bluetooth використовує діапазон ISM (англ. Industry, Science and Medicine), який є вільним від ліцензування та широко використовується в побутових приладах та бездротових мережах. Цей діапазон знаходиться в діапазоні 2,4-2,4835 Ггц. Сигнал Bluetooth формується за допомогою методу FHSS (Frequency Hopping Spread Spectrum - перехоплення частотного діапазону з псевдовипадковою перебудовою). Метод FHSS є простим у використанні та забезпечує стійкість до широкосмугових перешкод, а також є недорогим у використанні [5].

Згідно з протоколом FHSS, Bluetooth використовує стрибкоподібну зміну частоти сигналу, яка змінюється 1600 разів в секунду. Для з'єднань доступно 79 робочих частот шириною 1 МГц, а в Японії, Франції і Іспанії це число складає 23 частотних канали. Послідовність перемикання між цими частотами псевдовипадкова і відома тільки передавачу і приймачу. Кожні 625 мкс (один часовий слот), передавач і приймач синхронно перебудовуються з однієї частоти на іншу. Таким чином, коли поряд працюють декілька пар

приймач-передавач, то вони не заважають один одному. Цей алгоритм є складовою частиною системи захисту конфіденційності інформації, яка передається.

Bluetooth підтримує різні схеми кодування для передачі цифрових даних і аудіосигналу. У разі втрати пакету інформації, цифрові дані будуть передані повторно, а аудіосигнал не повторюється (як правило). Це забезпечує передачу даних зі швидкостями 723,2 Кбіт/с зі зворотним каналом 57,6 Кбіт/с, або 433,9 Кбіт/с в обох напрямах. Bluetooth підтримує дуплексний режим з часовим розділенням (TDD) для повнодуплексної передачі. Також підтримується ізохронна і асинхронна передача даних і проста інтеграція з TCP / IP. Щоб забезпечити низьке енергоспоживання, пристрої Bluetooth повинні мати потужність не більше 0,1 Вт. Кожен пристрій має унікальну 48-бітову мережеву адресу, яка сумісна з форматом стандарту локальних мереж IEEE 802.

1.4. Основні терміни технології Bluetooth.

Технологія Bluetooth включає в себе багато специфічних особливих термінів, які використовуються для опису принципів роботи та специфікації, а саме:

- "Пікомережа" або "Піконет" - це набір пристроїв, які з'єднані між собою за допомогою Bluetooth технології в спеціальний спосіб. Починаючи з двох з'єднаних пристроїв, таких як портативний ПК та стільниковий телефон, піконет може розширюватися до восьми з'єднаних пристроїв. Адресний простір обмежений 3 бітами. Усі Bluetooth пристрої мають ідентичну реалізацію і є одноранговими, але при установці піконет один пристрій діє як майстер для синхронізації, а інші - як слейв для підтримки піконет-з'єднання [6].
- "Мережа розкиду" або "Scatternet" - це дві або більше незалежних і несинхронізованих мереж піконет, які взаємодіють між собою. Процес

встановлення з'єднання відбувається за допомогою пристроїв-майстрів та пристроїв-слейвів, які можуть діяти як майстри та слейви в різних мережах піконет [5].

- "Пристрій-майстер" - це пристрій в піконет, який генерує синхронізуючі імпульси та послідовність стрибків для синхронізації інших пристроїв в піконет [6].
- "Пристрій-слейв" - це будь-який пристрій в піконет, який не є майстром і може бути до 7 активних пристроїв на кожен пристрій-майстер [6].
- "MAC-адреса" - це 3-бітний адрес Media Access Control, який використовується для розрізнення пристроїв, підключених до піконет.
- Парковані пристрої не мають MAC-адрес, але синхронізовані в піконет."
- "Sniff Mode і Hold Mode - це режими збереження енергії для пристроїв в піконеті, в яких активність знижена."
- "Ad-hoc мережа - це мережа, в якій взаємодія встановлюється між комплексними станціями без використання точки доступу або сервера."
- "Link Manager Protocol - це протокол, що використовує повідомлення LMP для налаштування зв'язку, безпеки та управління."
- "Інтерференція сигналів або Завади - це будь-яке змінення або пошкодження інформації, що передається за допомогою сигналів від передавача до приймача в каналі зв'язку, наприклад, сонячна інтерференція в супутниковому зв'язку."

Також в термінології Bluetooth присутні різні користувацькі моделі.

Користувацькі моделі описують, як можна використовувати пристрої Bluetooth, і кожна з них містить один або більше профілів, що визначають рівні протоколів і функції, які потрібно використовувати.

Internet Bridge. Ці моделі описують, як можна використовувати пристрої Bluetooth, і кожна з них містить один або більше профілів, що визначають рівні протоколів і функції, які потрібно використовувати.

Three-in-One Phone. Користувацька модель Three-in-One Phone описує можливість телефонного апарату приєднуватися до трьох різних провайдерів мобільних послуг. Телефон може функціонувати як бездротовий телефон у громадській комутованій телефонній мережі в домашніх умовах за фіксовану оплату оператора зв'язку. Цей сценарій включає здійснення дзвінків через голосову базову станцію та прямі дзвінки між двома терміналами через базову станцію. Також телефон може працювати як "walkie-talkie" або як додатковий пристрій телефону без додаткової оплати, приєднуючись безпосередньо до інших телефонів. Нарешті, модель Three-in-One Phone може працювати як стільниковий телефон, підключаючись до стільникової інфраструктури. Обидва бездротові сценарії використовують одну і ту ж мережу протоколів.

Ultimate Headset. Користувацька модель Ultimate Headset є бездротовою навушниковою трубкою з Bluetooth технологією, яка може працювати в умовах віддаленого аудіо-інтерфейсу вхідного та вихідного пристроїв, таких як мобільні телефони або ПК. Аналогічно до моделі Internet Bridge, вона вимагає використання протоколів, які складаються з двох частин: одна для команд AT, щоб керувати передачею необхідних даних, наприклад мовлення, з мобільного телефону або іншого пристрою, а інша - для контролю телефонів, наприклад, щодо процесу відповіді та переривання дзвінків.

LAN Access. Користувацька модель LAN Access подібна до користувацької моделі Internet Bridge, але має певні відмінності. Вона не потребує протоколів для команд AT, і її опис полягає в тому, як термінали введення даних використовують місце доступу LAN як бездротове з'єднання з локальною мережею. Під час встановлення з'єднання термінали введення даних працюють так, ніби вони були підключені до LAN за допомогою з'єднання dial-up [7].

File Transfer. File Transfer дозволяє передавати дані між різними пристроями Bluetooth, включаючи файли, папки, директорії і потоки даних в

різних форматах. Крім того, вона дозволяє переглядати вміст папок на віддалених пристроях та обмінюватись візитними картками за допомогою формату vCard. Модель File Transfer базується на протоколі GOEP.

Synchronisation. Користувацька модель Synchronisation дозволяє автоматично синхронізувати дані між різними пристроями, такими як настільний ПК, портативний ПК, мобільний телефон та ноутбук. Ця модель передає візитні картки, список завдань, записи з щоденника та інші дані, щоб їх можна було легко працювати з ними на комп'ютері, стільниковому телефоні або КПК, використовуючи звичайні протоколи та формати.

1.5. Стек протоколів Bluetooth.

Стек протоколу Bluetooth - це набір програмного забезпечення, яке забезпечує реалізацію протоколу Bluetooth на пристроях. Він складається з різних рівнів, кожен з яких відповідає за певні функції. Нижній рівень стеку протоколу відповідає за роботу з апаратною частиною, такою як передача інформації через радіоінтерфейс. Середній рівень забезпечує з'єднання між пристроями, управління передачею даних та контроль помилок. Верхній рівень стеку протоколу Bluetooth відповідає за забезпечення різних додаткових функцій, таких як передача голосової інформації та збереження контактів. Загалом, стек протоколу Bluetooth забезпечує можливість бездротового з'єднання між різними пристроями, що підтримують Bluetooth, і передачу даних між ними [8].

На рис. 1.1 зображено стек протоколів технології Bluetooth.

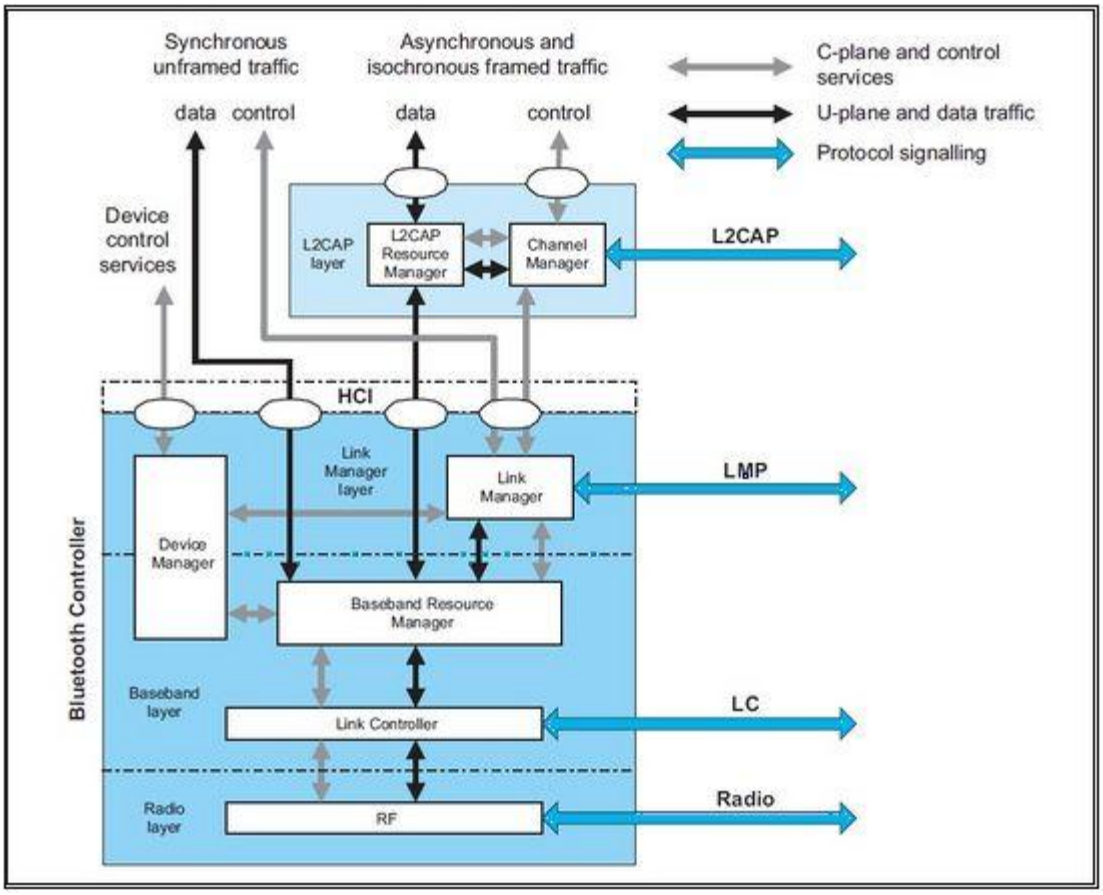

Рис. 1.1. Стек протоколів технології Bluetooth

З рис. 1.1. видно, що стек протоколу Bluetooth складається з рівнів, починаючи з рівня радіо на найнижчому рівні, який формує фізичний інтерфейс з'єднання. Рівень протоколу зв'язку baseband і Link Manager Protocol (LMP) встановлюють та контролюють з'єднання між пристроями Bluetooth. Ці три рівні підтримуються обладнанням або програмним забезпеченням. Рівень Host Controller є необхідним, щоб з'єднати Bluetooth з верхнім протоколом-L2CAP (Logical Link Control і Adaptation Protocol) за допомогою інтерфейсу. Провідний контролер є необов'язковим, якщо L2CAP знаходиться в програмному забезпеченні на хості. Якщо L2CAP також знаходиться на модулі Bluetooth, цей рівень не потрібен, оскільки L2CAP може безпосередньо зв'язуватися з LMP і baseband. Програми постійно знаходяться на рівнях вище L2CAP.

Рівні зв’язку. Цей зв'язок використовує діапазон ISM, що знаходиться на частоті близько 2.4 ГГц та використовує поширення спектру

для передачі даних. В більшості країн цей діапазон простирається від 2400 до 2483.5 МГц та використовується для оптимізації поширення спектру. Однак, деякі країни використовують менший діапазон ISM та використовують нижню шкалу. Техніка frequency hopping (FH) використовується для поширення спектру, оскільки у цьому діапазоні можуть бути нескоординовані мережі, що заважають його роботі. Швидкі FH та короткі передачі даних використовуються для зменшення відсотка помилок, зокрема через втручання від мікрохвильових печей, які працюють у цій частоті. CVSD-кодування адаптоване для передачі голосу, що може призводити до високого відсотка помилок. Заголовки пакетів захищені спеціальною схемою корекції помилок для підвищення стійкості проти збоїв. У зв'язку з цим, переходи по частоті фіксуються на 2402 k МГц, де k = 0,1 ,..., 78, а номінальна частота переходу становить 1600 стрибків в секунду [3].

Baseband.

Baseband - це рівень протоколу, який контролює зв'язок. Він відповідає за послідовність перельоту частоти та кодування рівня для забезпечення безпеки з'єднань. Існують два типи з'єднань: Синхронне орієнтоване з'єднання (SCO), яке призначене для передачі синхронних даних, таких як голос, та Асинхронне з'єднання (ACL), яке використовується для передачі даних, які не потребують синхронного зв'язку.

Baseband забезпечує необхідні можливості для синхронізації годинника та встановлення з'єднань між пристроями. Він також містить процедури запиту для виявлення пристроїв, які знаходяться поблизу. Для виправлення помилок у пакетах використовуються різні типи пакетів, які відрізняються за ємністю та витратами на виправлення помилок. Для інформаційного контролю та управління з'єднанням передбачено п'ять різних типів каналу, а також кілька функцій для генерації клавіш кодування та з'єднання.

Протокол менеджера зв’язку. Основні функції LMP можна класифікувати:

1. Управління мережею Piconet.
2. Конфігурація з’єднань.
3. Функції безпеки.

Piconet представляє собою групу пристроїв, що з'єднані з загальним каналом, який ідентифікується за його унікальною послідовністю "перельоту".

Фундаментальними "будівельними блоками" топографії Bluetooth є пристрої майстер і слейв, де пристрій майстер у піконеті забезпечує генерацію синхронізуючих імпульсів та послідовність стрибків для синхронізації всіх інших пристроїв у піконеті. Процес підключення контролюється майстером, який називається "master", а до нього може бути активно приєднано до семи пристроїв. Крім того, більше пристроїв може бути підключено в стані "parked" з низьким енергоспоживанням. Для з'єднання пристроїв у мережі piconet можна використовувати SCO або ACL. Канал управляється майстером, за допомогою Lin Manager в кожному пристрої, і будь-які два або більше пристроїв для з'єднання повинні встановити між собою мережу piconet. Крім того, кожен пристрій може одночасно належати декільком мережам. Майстер також визначає шаблон, на якому працюють всі слейв-пристрої його мережі піконет і синхронізує їхню роботу [8].

На рис.1.2. зображено, що кожен пристрій може одночасно належати до двох мереж (Piconet і Scatternet).

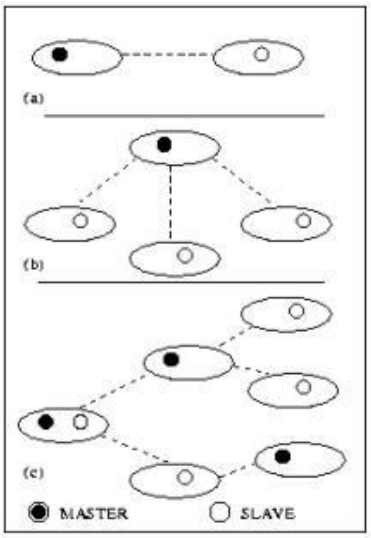

Рис.1.2. Piconet і Scatternet

З рис.1.2. видно, що стандарт Bluetooth дозволяє створювати мережу scatternet з піконетами (до 10), які можуть бути незалежними та не синхронізованими між собою. Щоб з'єднати ці піконети в мережу scatternet, необхідно мати принаймні один загальний пристрій, який буде виступати як майстер у одному піконеті та як слейв в іншому. За допомогою інтерфейсу Bluetooth можна з'єднати до 71 пристрою в межах окремої scatternet, але використання пристроїв-гейтів дозволяє забезпечити зв'язок на більшій відстані через Інтернет.

a) Piconet включає в себе два пристрої (мастер та слейв), або один майстер та один слейв.

b) Piconet може містити кілька пристроїв (мастер і до 7 слейвів).

c) Scatternet - це комбінація кількох мереж Piconet, яка дозволяє з'єднувати до 10 мереж разом.

LMP забезпечує можливість підключення/відключення пристроїв-слейвів, обмін функціями між мастер- та слейв-пристроями, а також з'єднання ACL/SCO. Крім того, LMP підтримує різні режими низького енергоспоживання, такі як hold, sniff та park, які дозволяють економно використовувати енергію, коли пристрої не передають дані.

Задачі налаштування з'єднання включають в себе параметри конфігурації з'єднання, контроль якості обслуговування та управління

потужністю, якщо це можливо на пристрої. Крім того, LMP виконує ідентифікацію пристроїв, які будуть з'єднані, і управління клавішами з'єднання.

Логічний контроль зв’язку і адаптивний протокол. Більшість додатків взаємодіють з цим протоколом, якщо ведучий контролер не використовується. Основні функції протоколу L2CAP:

- Мультіорганізація. Протокол мультіорганізації повинен забезпечувати можливість одночасного використання з'єднання між двома пристроями декількома додатками.

Сегментація і повторне об’єднання. Метою протоколу є зменшення розміру пакетів, які надаються додатками, до розміру пакетів, які приймаються рівнем baseband. Хоча L2CAP може приймати пакети розміром до 64Кб, базові пакети можуть бути прийняті до 2745 біт. Після отримання таких пакетів, L2CAP виконує процедуру зворотного об'єднання сегментованих пакетів у правильному порядку.

Якість обслуговування. L2CAP дозволяє додаткам встановлювати QoS на певних параметрах, таких як максимальна пропускна здатність, час очікування та затримка. Основна функція L2CAP полягає в забезпеченні мережевих функцій для додатків та більш високорівневих протоколів [1].

Інтерфейс головного контролера. Основна структура, що показує, як рівні головного контролера розташовані всередині стека протоколу n.

На рис. 1.3. зображено провідний контролер в стеку протоколу.

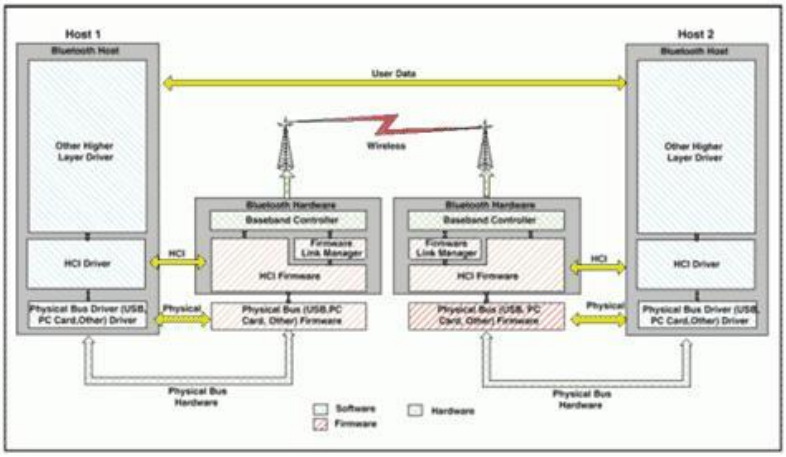

Рис.1.3. Провідний контролер в стеку протоколу

З рис. 1.3. видно, що для більшості пристроїв, підтримка Bluetooth може бути додана в якості розширювальної плати, наприклад, в ПК або ноутбук, апаратні засоби Bluetooth можуть бути додані як PCI-карта або USB-адаптер. Зазвичай, апаратні модулі здійснюють нижчі радіо рівні, такі як baseband і LMP. Дані, які будуть передані LMP і baseband, потім проходять через фізичну шину, наприклад, USB. Хост, яким є ПК, має мати драйвер для цієї шини, а картка Bluetooth потребує "інтерфейс контролера хоста", щоб прийняти дані з фізичної шини. Таким чином, для роботи потрібні додаткові рівні [3].

HCI драйвер. Цей драйвер форматує дані, які будуть передані від контролера хоста на апаратні засоби Bluetooth через інтерфейс, що знаходиться вище фізичної шини.

Інтерфейс головного контролера. Розміщується на апаратних засобах Bluetooth і підтримує зв'язок поверх фізичної шини.

Рівень програми. L2CAP доступний для додатків безпосередньо або через протоколи, такі як RFCOMM, TCS і SDP. Додатки можуть використовувати інші протоколи, такі як TCP-IP або WAP. Додатки можуть запускати інші протоколи, такі як PPP, FTP, або інші, якщо це необхідно для додатку. Для перевірки сервісних можливостей пристроїв, що доступні в зоні

дії, програма може використовувати SDP. Багато моделей використання були запропоновані виробниками, серед яких:

1. Три в одному: Телефонна трубка може виконувати функцію селекторного зв'язку в офісі без додаткової плати за користування телефоном. Крім того, користувач може в будь-який момент використовувати будь-який з трьох режимів: як селекторний зв'язок, PSTN або мобільний телефон.

2. «Портфельна хитрість»: RF-зв'язок забезпечує з'єднання між пристроями без необхідності у прямій видимості між ними. Це означає, що навіть якщо ноутбук знаходиться в портфелі, мобільний телефон може підключатися до нього та користуватися його можливостями, наприклад, електронною поштою.

3. Автоматичний синхронізатор: забезпечення бездротового зв'язку між PDA користувача, портативної ЕОМ і мобільним телефоном дозволяє додаткам автоматично оновлювати та синхронізувати дані, якщо зміни внесені на одному пристрої.

4. Бездротові навушники та гарнітури: ці бездротові навушники (гарнітури) дають користувачам доступ до їх мобільних пристроїв та навіть аудіо, незалежно від того, де пристрої зберігаються, наприклад, в кишені. Це дозволяє здійснювати операції hands-free.

5. Автомобільні комплекти: Hands-free пристрої дозволяють водіям залишатися на зв'язку, не відволікаючись від керування автомобілем [8].

На рис. 1.4. зображено огляд протоколів Bluetooth.

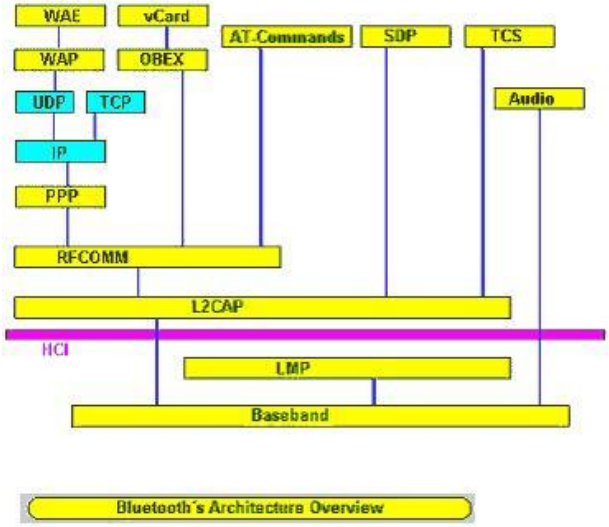

Рис. 1.4. Огляд протоколів Bluetooth

З рис.1.4. видно огляд протоколів, що використовуються і підтримуваних Bluetooth, та показано, як вони взаємодіють між собою. Як зазвичай у такого роду схемах, протоколи зображені відповідно до їх приблизної ієрархії, де "вищі" протоколи розташовані вгорі. З цієї ієрархії можна зробити два висновки щодо обміну даними.

1. Протоколи на вищих рівнях залежать від тих, що розташовані на нижчих рівнях, проте нижчі протоколи можуть працювати самостійно або підтримувати інші протоколи на вищих рівнях.

2. Протоколи на вищих рівнях зазвичай більш пристосовані до потреб користувача, оскільки забезпечують послуги, що орієнтовані на людину.

Технологія Bluetooth зазвичай використовується для комунікацій на близьких відстанях, але за допомогою спеціальних технічних засобів можна розширити дальність взаємодії пристроїв до 100 метрів. Для досягнення такої дальності необхідно забезпечити вихідну потужність передавача в 100 мВт, яка відповідає "Class 1" в специфікації Bluetooth ver. 1.0, ref. 2. Це означає, що передавач має можливість регулювати потужність для зменшення впливу перешкод, а також враховувати рівень фонового шуму при чутливості приймача -70 дБм [8].

На рис. 1.5. зображено приймально-передавальний пристрій Bluetooth.

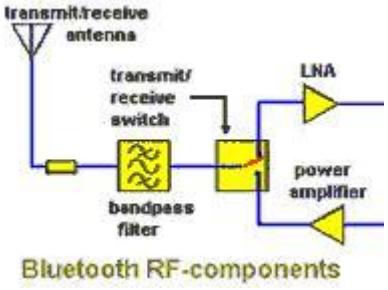

Рис.1.5. Приймально-передавальний пристрій Bluetooth

З рис. 1.5. можна бачити, що в RF-частини прийомопередавача Bluetooth є:

1. Існує смуга пропускання, яка фільтрує всі частоти, що перевищують 1 Мгц, що використовується одночасно для передачі та прийому.
2. Перемикач передавання-прийому, який переключає підсилювач потужності до антени під час передачі та LNA до антени під час прийому.
3. LNA або малошумний підсилювач, який знижує рівень шуму від антени до передавача/приймача.

RF-частина повинна відповідати вимогам, встановленим у специфікації. Для її роботи необхідно джерело живлення від 2.7 до 5 вольт, що працює в діапазоні від -20 до +60 С, з ефективністю принаймні 30% при 140 мА. Під час прийому вихідна потужність має бути вимкнена. Продукт, що задовольняє ці вимоги, називається "MAX2240". Для установки змінного посилення використовуються два цифрових керуючих біти, що дозволяють отримати 4 окремих рівня вихідної потужності.

1.6. Установка з’єднання в Bluetooth

Розглянемо основні процедури, які повинні бути виконані пристроями Bluetooth, щоб встановити між ними з’єднання. У випадку, коли потрібно здійснити з’єднання, пристрій автоматично виконав б наступні кроки (крім

кроку реєстрації, актуальної, якщо пристрій вперше використовується в цьому середовищі):

1. Запит: при переході до нового середовища пристрій автоматично ініціалізує запит, щоб дізнатися, які пристрої доступні в його радіусі дії. Це призведе до наступного:

a) Всі доступні пристрої відповідають на запит.

b) Пристрій вибере один з доступних пристроїв.

2. Пейджинг: пристрій викликає процедуру пейджингу для baseband, що в результаті синхронізує пристрій з точкою доступу та проводить інші необхідні ініціалізації.

3. Встановлення з'єднання: LMP встановлює з'єднання з точкою доступу. Оскільки додаток в даному випадку поштовий (email), буде використано ACL з'єднання.

4. Сервіс: LMP використовує протокол SDP для встановлення доступного сервісу, зокрема поштового сервісу, або для звернення до іншого хоста. Припустимо, що сервіс доступний, інакше програма не зможе продовжувати діяти. Інформація щодо інших сервісів також може бути представлена користувачеві.

5. За допомогою інформації, отриманої від SDP, пристрій зможе створити канал L2CAP до пункту доступу, який може бути використаний додатком або іншим протоколом, таким як RFCOMM, безпосередньо.

6. При необхідності поштової програми RFCOMM або іншого додатку буде створено канал відповідно до L2CAP каналу. Це дозволяє використовувати додатки, розроблені для послідовних портів, без модифікації для Bluetooth-платформи.

7. Забезпечення безпеки: Якщо пункт доступу обмежує доступ до певної кількості користувачів або пропонує безпечне з'єднання для зареєстрованих користувачів, тоді пункт доступу запропонує запит на безпеку під час встановлення з'єднання. Користувач повинен знати правильний PIN-код для доступу до сервісу, але цей PIN-код не передається

через бездротовий канал. PIN-код генерується для використання в інших кодах, тому дуже складно його підібрати. Якщо використовується безпечний режим, буде встановлено кодування передачі.

8. PPP: Оскільки PPP-з'єднання використовується з послідовного модему, як у випадку з dial-up, те ж саме додаток тепер зможе запустити PPP через RFCOMM, використовуючи емульований послідовний порт. Це з'єднання дозволить користувачеві отримати доступ до його поштової скриньки та ін.

9. Мережеві протоколи: Мережеві протоколи, такі як TCP/IP, IPX та Appletalk, можуть передавати та отримувати дані через канал без будь-яких труднощів [3].

Користувач потребує взаємодії тільки на початковому етапі, коли необхідно ввести свій логін для входу в систему електронної пошти та для забезпечення додаткової безпеки. Після цього всі інші кроки здійснюються автоматично. Описані процедури детально розглянуто, щоб продемонструвати процес налаштування підключення.

Годинник. Кожен модуль Bluetooth включає в себе вбудовану систему часу, яка визначає час і частотні передачі. Годинники Bluetooth не потребують регулювання і не вимикаються, але їхні зміщення використовуються для синхронізації з іншими модулями. Годинники Bluetooth не впливають на час доби, але вони дуже важливі для трансівера Bluetooth, тому що вони синхронізують безліч важливих подій, необхідних для забезпечення зв'язку. Одиницею часу є принаймні половина довжини слоту TX або RX або 312,5 мікросекунд. Годинники мають денний цикл, і якщо вони обладнані лічильником, 28-бітний лічильник вимагає обертання близько 228 разів. Мережа piconet синхронізується з годинами "майстра", і підпорядковані пристрої зберігають необхідне значення зміщення для синхронізації каналу. Мінімальна точність годин становить -/-20 ppm в активному режимі і/-250ppm в режимі малої активності, такому як Hold, Sniff, Standby і Park.

Запит на пейджинг. Це перші кроки в процесі встановлення з'єднання. Зазвичай пристрій за замовчуванням перебуває в режимі Standby, при якому споживання енергії дуже низьке, і рідні годинники продовжують роботу. Але є можливість вийти з цього режиму і перейти в режими Inquiry, Inquiry Scan, Inquiry Response, Page або Page Scan для подальшого встановлення з'єднання.

Встановлення зв’язку

Коли пристрій переходить у стан зв'язку, LMP може розпочати процес встановлення з'єднання. З'єднання L2CAP ґрунтуються на концепції каналів, які ідентифікуються за допомогою ідентифікаторів каналу, аналогічних сокетам в TCP/IP. Канал, відмінний від каналу Piconet, характеризується адресою пристрою, з яким встановлюється двостороннє з'єднання, та ідентифікатором каналу. Загальний процес встановлення з'єднання може бути описаний наступними кроками:

1. Пакети POLL та відповіді використовуються для передачі конфігураційної інформації без необхідності взаємодії з хостом. Ці дії входять до процедури парування, що є зв'язком двох або більше пристроїв з метою створення спільного секретного ключа Kinit для подальшого використання при зв'язку. Термін "підгонка пари" може також використовуватися для позначення процедури.

2. Для початку процедури зв'язку, на обох пристроях потрібно ввести PIN-код. Наприклад, коли дві людини бажають сполучити свої телефони, вони можуть домовитися про використання одного PIN-коду.

3. Якщо запитуваний пристрій не може або не бажає відповісти, він надсилає пакет LMP_NOT_ACCEPTED, інакше надсилається пакет LMP_ACCEPTED. Для цього прикладу ми будемо розглядати два пристрої - A та B. Перший пристрій, A, є головним, а другий, B - веденим. Створення ключа Kinit починається після введення PIN-кодів.

4. Якщо необхідно, пристрій може запросити відключення ролі. Перший пристрій відповідає пакетом для прийняття або відхилення запиту. Зв'язок встановлюється на рівні менеджера зв'язку. Додаток може не знати,

які послуги доступні, тому він може використовувати SDP для виявлення цих послуг [9].

SDP

Часті зміни у середовищі Bluetooth вимагають виявлення доступних послуг в полі зору, що забезпечує SDP. Цей протокол надає програмні засоби для виявлення доступних сервісів та їх характеристик, як описано в основних специфікаціях.

Щоб виявити послуги інших пристроїв, пристрій Bluetooth запускає SDP-сервер, а клієнт запускається для кожної програми. Один пристрій може запустити тільки один сервер SDP, який обслуговує записи кожної служби для виявлення пристроєм. Клієнт може надсилати запит на сервер, щоб шукати сервіси за специфічним класом або переглядати всі класи доступних сервісів. Якщо сервер має тільки декілька сервісів, вони можуть не бути розділені на класи і описи сервісів не відправляються пристрою. Інакше, класові описи відправляються, щоб клієнт міг дізнатися деталі в межах класу.

Знайдені сервіси можна доступатися через інші протоколи, засновані на L2CAP. L2CAP встановлює з'єднання з пристроями на основі ідентифікаторів каналу, які ідентифікують канали, подібні до роз'ємів в IP TCP. Кожен канал приймається до заповнення дуплексу і може бути двухточковим або багатоточковим. Запити від нижніх рівнів щодо з'єднань, необхідних програмами на віддалених пристроях, також обробляються L2CAP відповідно до залучених додатком.

SCO з'єднання передають свої дані безпосередньо через Baseband, а не по L2CAP. Однак, L2CAP створює окремий сигнальний канал для запитів з'єднання, конфігурації, роз'єднання та інших операцій (для тестування). Пакети L2CAP не містять CRC або інших перевірок помилок, але покладаються на смугу baseband для захисту даних та забезпечення впорядкованої доставки.

Інтеракція цього протоколу з верхніми та нижніми рівнями поділяється на події та дії. Події охоплюють всі повідомлення, які L2CAP

отримує від більш низьких або вищих рівнів, тоді як дії відповідають на ці події. Нижчим рівнем може бути LMP або HCI, тоді як вищим рівнем може бути будь-який додаток. Типова послідовність подій та дій для встановлення з'єднання може мати наступний порядок:

1. Подія та Дія 0: При отриманні запиту з'єднання від вищого рівня, L2CAP посилає пакет запиту на віддалений пристрій через baseband.
2. Подія та Дія 1: Віддалений пристрій L2CAP отримує запит та відповідає підтвердженням з'єднання, попередньо взаємодіючи з відповідним додатком для перевірки обробки запиту.
3. Подія та Дія 2: Локальний пристрій L2CAP отримує підтвердження з'єднання та ініціює запит конфігурації, вказуючи параметри, такі як максимальний модуль payload та межі часу очікування.
4. Подія та Дія 3: Віддалений пристрій L2CAP отримує запит конфігурації та відповідає на нього зі своєю конфігурацією, а також може надіслати запит на додаткові параметри конфігурації.
5. Подія та Дія 4: Локальний пристрій отримує відповідь на запит конфігурації та надсилає відповідь із своєю конфігурацією.

L2CAP виступає ініціатором на місцевому пристрої, а адресатом є L2CAP на пункті доступу або на іншому пристрої Bluetooth, з яким він взаємодіє. Важливо зазначити, що стрілки, що вказують до ініціатора або адресата, вказують на події для L2CAP, тоді як стрілки, що вказують за межі дії, вказують на зв'язки між двома L2CAP на різних пристроях. LP Менеджери Зв'язки на цих пристроях зображені вертикальними рядками. Назви, що починаються з WA, позначають з'єднання з додатком більш високого рівня, для якого встановлюється канал.

Відкритий стан (OPEN) відображає інтервал з'єднання додатків. Останні кроки у таблиці відносяться до роз'єднання. За цим можуть бути передані дані програми або можуть бути виконані процедури захисту.

Зв’язок додатків

Після успішного встановлення з'єднання Bluetooth прикладні дані зможуть бути передані. Для цього потрібно запустити протокол L2CAP на більш високому рівні. В системі Bluetooth доступні три протоколи:

RFCOMM – емуляція послідовног порту через бездротове з’єднання;

SDP – Service Discovery Protocol, який допомогає пристроям виявити доступні послуги поблизу

TCS – це Telephony Control Protocol Specification, описує, як управляти запитами і передавати голосові сигнали через Bluetooth. Додатки користувачів та інші механізми доступу до мережі, такі як IP TCP, PPP, IrDA OBEX, WAP та HomeRF, можуть бути використані на рівні L2CAP або трьох вищезгаданих протоколах, якщо додаток вибере їхні послуги.

Якщо підключення не було зафіксовано протягом часу запуску програми, додаток повідомляє, що більше не потрібно підключення. LMP відправляє без відповіді пакет LMP_detach на віддалений пристрій, після чого відбувається роз'єднання.

Отже, в ході проведення аналізу було виявлено, що перевагами технології Bluetooth є:

- Низьке споживання енергії, що дозволяє використовувати Bluetooth-пристрої довше без підзарядки батареї
- Доступність: Багато пристроїв підтримують Bluetooth, що робить його досить універсальним і зручним для використання.

Недоліки технології Bluetooth:

- Обмежена дальність передачі даних: зв’язок між Bluetooth-пристроями може бути досить слабким, особливо, якщо є перешкоди.
- Низька швидкість передачі даних: порівняно з іншими бездротовими технологіями, такими як Wi-Fi або NFC, швидкість передачі даних Bluetooth може бути доволі повільнішою.

- Підвищена вразливість до атак: на жаль, Bluetooth може бути піддано різним атакам. Типи атак, до яких вразливі Bluetooth-пристрої будуть розглянуті в наступному розділі.

РОЗДІЛ 2 . БЕЗПЕКА BLUETOOTH

2.1. Атаки на Bluetooth

Існує велика кількість вразливостей у технології Bluetooth , і багато рівнів захисту можуть бути легко зламаними. Крім того, більшість користувачів не усвідомлюють реальних загроз, пов'язаних з можливими атаками на їх пристрої з використанням технології Bluetooth, і тому не вживають достатніх заходів для захисту своїх пристроїв .

2.1.1. Злам PIN-коду

В 2006 року Авіша Вул та Янів Шакед опублікували статтю, яка містить докладний опис можливих атак на Bluetooth-пристрої. У статті було розглянуто як активні, так і пасивні методи атаки, які дозволяють зловмисникам отримати PIN-код пристрою та підключитись до нього. Пасивна атака полягає в тому, що зловмисник може прослуховувати процес ініціалізації з'єднання, отримувати та аналізувати дані, які надходять під час цього процесу та використовувати їх для встановлення з'єднання (spoofing) [10].

Атака на Bluetooth-пристрій полягає у визначенні PIN-коду, який захищає з'єднання. Зловмисники проводять таку атаку для того, щоб отримати доступ до всіх зашифрованих повідомлень, що передаються через Bluetooth, і обійти процедуру аутентифікації, щоб мати доступ до пристрою. Існує кілька способів проведення подібної атаки, один з яких використовує людський фактор, а інший - математичні методи.

Після успішної реалізації такої атаки зловмисник зможе перехоплювати всі передані дані між Bluetooth-пристроями, включаючи зашифровані повідомлення. Навіть якщо трафік був зашифрований, зловмисник може прослуховувати отримання PIN-коду.

2.1.2. Атака з підміною пристрою

Атакуючи, зловмисник замінює налаштування вже авторизованого Bluetooth-пристрою. Для цього атакуючий налаштовує новий пристрій з такою самою адресою, списком доступних профілів та протоколом роботи, як і в автентичному пристрої [11].

Ця заміна може бути здійснена через наявність Bluetooth-пристроїв, що дозволяють змінювати адресу. В результаті цієї атаки зловмисник може здійснювати AT-команди на телефоні та мати доступ до будь-яких файлів на ньому.

2.1.3. Атака на piconet-мережу

Ця атака спрямована на руйнування мережі piconet за допомогою пристрою, який не є частиною цієї мережі. Вона базується на особливостях побудови Bluetooth-мережі, згідно з якими пристрій типу Master може підтримувати кілька з'єднань для створення розширених мереж (scatternet).

Атакуючий може виконати описану вище атаку, підмінивши пристрій на один з пристроїв piconet-мережі, що призведе до звернення підміненого пристрою до пристрою типу Master. Таке поведінка пристрою типу Slave може спричинити втрату контролю над piconet-мережею пристроєм типу Master та руйнування встановлених в piconet-мережі зв'язків. Наявність такої вразливості не пов'язана з виробником конкретних пристроїв Bluetooth, а є загальним недоліком у побудові пристроїв цього типу [12].

2.1.4. Атака зі скиданням ключа зв'язку

Дана атака, примушує пристрій Bluetooth видалити збережений ключ зв'язку, що дозволяє зловмиснику перехопити обмін ключами. Останнім часом цю атаку можна проводити в режимі реального часу, тому зловмисник може спровокувати обмін ключами у потрібний для нього момент.

Атакуючий, щоб провести атаку, повинен знати адреси пристроїв, що приєднані до мережі. Він підробляє адресу одного з пристроїв і з'єднується з іншими. Оскільки зловмисник не має ключа зв'язку, коли цільовий пристрій посилає запити на автентифікацію, пристрій атакуючого відповідатиме «HCI_Link_Key_Request_Negative_Reply». Це може привести до скидання ключа зв'язку в цільовому пристрої та переходу його в режим з'єднання. Зазначимо, що ця атака може бути проведена в режимі реального часу, що дозволяє зловмиснику викликати запити на автентифікацію в необхідний момент [12].

2.1.4. Атака підробки точки доступу

Деякі мобільні телефони мають вразливість, яка дозволяє зловмиснику перехоплювати весь вихідний трафік з пристрою. Це стає можливим через те, що деякі телефони відображають пристрої, виявлені під час пошуку, за іменами, які були присвоєні їм їх власниками. Ці імена можуть повторюватися, і користувач може легко сплутати один пристрій з іншим, що створює можливість для зловмисника перехоплювати трафік.

Таким чином, зловмисник може виконати атаку, замінивши пристрій, та перехоплювати весь вихідний трафік користувача. Ця атака може мати кілька застосувань. Наприклад, існують послуги, які дозволяють звернутися до точки доступу до Інтернету через Bluetooth. Якщо зловмисник дізнається PIN-код точки доступу, її адресу та ім'я (що можна здійснити за допомогою описаних вище атак), він зможе перехоплювати всю вихідну інформацію користувача [13].

Існує ще один сценарій атаки, який може мати наступний вигляд. Зловмисник може відправити користувачу повідомлення, що містить прохання про авторизацію від імені знайомого абонента, який насправді є зловмисником. Така ситуація може статися, наприклад, при використанні мобільного кіоску, де користувач може замовити та сплатити мелодію для свого мобільного телефону, а потім завантажити її через Bluetooth. Зловмисник може відправити запит авторизації через Bluetooth під іменем "mobile-kiosk" відразу після оплати, що може призвести до атаки на передані дані та передачу вірусу користувачу, який погодився на авторизацію.

2.1.5. Атака з розкриттям інформації про пристрій

Ця атака має на меті отримання повної інформації про пристрій, включаючи наявність вразливостей, вона спрямована на те, щоб обійти механізми захисту Bluetooth, які призначені для приховування всіх даних про пристрій. Існує кілька різних варіацій цієї атаки

Перший тип атак націлений на обхід режиму "прихованого" пристрою. Цей режим був розроблений розробниками Bluetooth, щоб користувач міг відкривати свою присутність лише тим користувачам, яких він знає, та залишатися непоміченим для сторонніх осіб, зокрема зловмисників. Завдяки цій атаці зловмисник може отримати адресу пристрою Bluetooth, який знаходиться в прихованому режимі. Другий тип атаки орієнтований на виявлення всієї інформації про пристрій. Ця атака дозволяє знайти модель пристрою, протоколи, якими він працює, та інші необхідні дані про пристрій, включаючи його вразливості.

Ця атака може використовуватись як додатковий інструмент для більшості атак на Bluetooth. Причина полягає в тому, що якщо зловмисник здійснює багато неуспішних спроб злому, то жертва може запідозрити, що її телефон став об'єктом атаки. Однак, якщо зловмисник має докладну інформацію про пристрій перед початком нападу, він зможе краще

спланувати свої дії та підібрати найбільш ефективний спосіб злому для цього пристрою [14].

2.1.6. Атака з використанням уразливих каналів

Ця атака дозволяє зловмиснику виконувати несанкціоновані дії з пристроями, що мають увімкнений Bluetooth.,

Ця проблема ґрунтується на вразливості деяких пристроїв, які мають передавач Bluetooth. Більшість таких пристроїв мають можливість віддаленого керування, наприклад, за допомогою гарнітури, але не передбачають авторизацію для підключення таких пристроїв. Це означає, що канал (channel) у профілі для гарнітури не захищено, хоча саме цим каналом гарнітура здійснює управління пристроєм, включаючи виконання AT-команд. Зловмисник може підключитися до такого каналу і виконувати неповноважні дії з пристроєм за допомогою віддаленого виконання AT-команд. Це дозволяє виконати такі дії: ініціювати телефонний дзвінок, надсилати SMS-повідомлення на будь-який номер, читати SMS з телефону, а також читати та записувати контакти телефонної книги, а також встановлювати переадресацію дзвінків [13].

Атаки з використанням цієї можливості можуть бути реалізовані різними способами, залежно від мети атакуючого.

2.1.7. Атака з переповненням буфера

Ця атака є DoS-атакою на пристрої з Bluetooth. Вразливі до неї можуть бути пристрої, які не перевіряють довжину пакета, що прийшов, і мають обмежену ємність буфера. Атакуючий може надіслати пакети досить великого розміру, які переповнюють буфер, що виділяється для них, тим самим заблоковуючи роботу пристрою. У результаті такої атаки мобільний телефон може вийти з ладу[13].

2.1.8. Атака з використанням уразливості OBEX (OPP)

Одна з найбільш ефективних атак Bluetooth, що базується на вразливості певних пристроїв, пов'язаній з недостатньою реалізацією аутентифікації при взаємодії між OBEX-клієнтом та OBEX-сервером. OPP (OBEX Push Profile) використовується для обміну візитними картками (vCard) та іншими об'єктами, і часто не потребує аутентифікації, що створює небезпеку для безпеки мобільних пристроїв. Зловмисники можуть скористатися цим недоліком і виконати запит OBEX GET до відомих файлів, наприклад telecom/pb.vcf (Адресна книга) або telecom/cal.vcs (календар), які можуть бути викрадені через відсутність аутентифікації [13].

2.1.10 Атака для визначення розташування об'єкта

Ця атака має на меті визначення місцезнаходження мобільного телефону. Для її здійснення зловмисник використовує дві вразливості протоколу Bluetooth, що дозволяють здійснювати наступні типи атак:

• Атака, що використовується відкриту адресу пристрою.

Адреса пристрою у протоколі Bluetooth передається у незашифрованому вигляді, що робить пристрій вразливим до атак на розкриття адреси. Більшість пристроїв Bluetooth не можуть змінюватися, що означає, що зловмисник може відстежувати розташування атакованої людини за її пристроєм, якщо він виявить адресу пристрою. Звичайно, радіус стеження обмежується кількома сотнями метрів, але за допомогою спеціального обладнання цей діапазон можна збільшити.

• Атака з використанням вразливостей автентифікації.

Деякі пристрої Bluetooth мають вразливість в реалізації автентифікації, через що зловмисник може здійснити атаку на мобільний телефон жертви і виконувати на ньому довільні AT-команди, включаючи

відправку SMS-повідомлень. Крім того, вище згадувалися атаки на уразливість OBEX, які можуть передавати на мобільний телефон файл з резидентною вірусною програмою, що також може виконувати AT-команди та надсилати повідомлення з розташуванням об'єкта [13].

2.1.11. Атака з регенерацією ключа

Атака з регенерацією ключа Bluetooth може призвести до створення несанкціонованого з'єднання з атакованим пристроєм. Ця вразливість ґрунтується на тому, що при встановленні авторизованого з'єднання між двома пристроями, якщо ключ зв'язку видаляється, зв'язок залишається активним. Власник атакованого пристрою може навіть не підозрювати, що з'єднання залишається активним, коли ключ зв'язку видаляється, що може бути потенційно небезпечним [12].

Після видалення ключа атакуючого, необхідно запитати регенерацію ключа. Якщо такий запит був успішним, зловмисник отримує новий ключ без необхідності автентифікації. Це дає зловмиснику можливість отримати несанкціонований доступ до пристрою до того моменту, поки ключ не буде видалений.

2.1.12. Атака з вразливістю в інтерпретації імені телефону

Вразливість, якою користуються при цій атаці, полягає в тому, що пристрій Bluetooth може виявитися у некоректному стані через неправильну обробку імені підключеного пристрою, що може викликати його збій. Ім'я пристрою Bluetooth зазвичай кодується у форматі UTF-8, який підтримується більшістю пристроїв на ринку. Але деякі програмні модулі можуть не перевіряти наявність у рядку імені будь-яких керуючих символів, що може призвести до зависання пристрою. Якщо зловмисник надішле таке ім'я до

цільового пристрою, це може викликати атаку, спрямовану на збій цього пристрою [15].

2.1.13. Атака з підробкою відправника

Атака може бути менш небезпечною з технічної точки зору, але вона може бути використана як засіб психологічного тиску. Її суть полягає в тому, що можна надіслати анонімне повідомлення на пристрій жертви, яке не можна відстежити, оскільки не містить зворотного адресу, що може створити дискомфорт та становити загрозу для психічного здоров'я жертви [15].

2.1.14. Атака з використанням уразливості RFCOMM

Протокол Bluetooth має вразливість на рівні RFCOMM, що дозволяє зловмиснику отримати доступ до пристрою жертви, підключившись до нього через сокет RFCOMM. Для цього необхідно спочатку створити канал зв'язку, а потім - підключитись до атакованого пристрою. Щоб передати команду на мобільний телефон жертви, зловмисник повинен здійснити запис у пристрій, використовуючи дескриптор файлу rfcomm_fp. Для проведення таких атак можна використовувати різноманітні допоміжні бібліотеки, наприклад, Bluez. Хоча атака не є дуже небезпечною, вона може стати джерелом психологічного тиску на жертву [15].

2.2. Вразливість Blueborne

Armis Labs відкрила новий спосіб атаки, який становить загрозу для основних операційних систем, включаючи Android, iOS, Windows і Linux, а також пристроїв Інтернету речей, які їх використовують. Цей спосіб атаки, який отримав назву «BlueBorne», поширюється через Bluetooth і використовує вісім пов’язаних уразливостей нульового дня, чотири з яких є

критичними. Зловмисники можуть використовувати BlueBorne для отримання контролю над пристроями, доступу до корпоративних даних і мереж, проникнення в безпечні мережі з «повітряним розривом» і поширення зловмисного програмного забезпечення на сусідні пристрої.

BlueBorne є вектором атаки, що дозволяє хакерам використовувати з'єднання Bluetooth для здійснення проникнення та повного контролю над цільовими пристроями, включаючи звичайні комп'ютери, мобільні телефони та все більшу кількість пристроїв IoT. Для здійснення атаки BlueBorne не потрібно, щоб цільовий пристрій було сполучено з пристроєм зловмисника або навіть було встановлено режим видимості [16].

Blueborne включає в себе 8 критичних вразливостей на таких платформах : Android, Linux, Windows.

2.2.1. Вразливості платформи Android

В операційній системі Android знайдено чотири вразливості, які стосуються всіх телефонів, планшетів та переносних пристроїв Android, за винятком тих, що використовують лише Bluetooth Low Energy. Зокрема, дві з них дозволяють виконувати код здалеку (CVE-2017-0781 і CVE-2017-0782), одна веде до витоку інформації (CVE-2017-0785), а остання дозволяє здійснювати атаку типу "Man-in-The-Middle" (CVE-2017-0783).

1. CVE-2017-0785 - це вразливість Android, яка може стати причиною витоку інформації. Ця вразливість існує в протоколі виявлення сервісів (Service Discovery Protocol), який дозволяє пристрою ідентифікувати інші Bluetooth-пристрої. Хакер може використовувати цю вразливість, щоб відправляти запити до сервера і отримувати доступ до байтів пам'яті, включаючи ключі шифрування [17].

2. CVE-2017-0781 - це вразливість в Android, що дозволяє виконання коду віддалено. Вона виявлена в Bluetooth Network Encapsulation Protocol (протоколі інкапсуляції мережі Bluetooth), який дозволяє

використовувати пристрій як модем для доступу до Інтернету. Ця уразливість може бути використана для порушення цілісності інформації в пам'яті і віддаленого виконання коду без взаємодії з користувачем [18].

3. CVE-2017-0782 (вразливість Android) дозволяє виконання коду віддалено. Вона знаходиться в профілі персональної мережі Bluetooth Network Encapsulation Protocol, який відповідає за встановлення з'єднання між пристроями. При атакуванні уразливості порушується цілісність інформації в пам'яті, що дає змогу виконати код віддалено [19].

4. CVE-2017-0783 - вразливість в Android, яка використовує атаку типу Man-in-The-Middle (Людина посередині). Ця уразливість виявлена в PAN-профілі стеку Bluetooth і дозволяє створювати шкідливий мережевий інтерфейс на цільовому пристрої, перенастроювати IP-маршрутизацію та примусово передавати повідомлення через цей інтерфейс. З цієї причини атака майже непомітна, оскільки вона не вимагає взаємодії з користувачем [20].

Від цих вразливостей постраждали такі пристроїв: Google Pixel, Samsung Galaxy, Samsung Galaxy Tab, LG Watch Sport, автомобільна аудіосистема Pumpkin.

2.2.2. Вразливості платформи Windows

Вразливість "Bluetooth Pineapple" (CVE-2017-8628) може бути використана зловмисником для здійснення атаки типу Man-in-The-Middle на всі комп'ютери з операційною системою Windows, що починаються з версії Windows Vista.

Уразливість CVE-2017-8628 в Windows, є аналогом CVE-2017-0783 і також дозволяє виконати атаку типу Man-in-The-Middle, використовуючи ті ж самі принципи при реалізації деяких протоколів Bluetooth [27].

2.2.3. Вразливості платформи Linux

Операційна система Linux є основою для багатьох пристроїв, а серед комерційних та споживчих платформ на її основі найбільш поширеною є ОС Tizen.

Усі пристрої з операційною системою Linux, що працюють під управлінням BlueZ, вразливі на витік інформації (CVE-2017-1000250). Крім того, всі пристрої з Linux версії 2.6.32 (випущеної в липні 2009 року) до версії 4.14 також піддаються ризику віддаленого виконання коду (CVE-2017-1000251).

1. Уразливість CVE-2017-1000250 в Linux (витік інформації) подібна до CVE-2017-0785 і присутня на сервері SDP (відповідає за автоматичне підключення пристроїв до служб, які надають інші пристрої). Ця уразливість дозволяє розкрити біт пам'яті і приводить до витоку ключів шифрування [21].

2. CVE-2017-1000251 (уразливість Linux. Переповнення стеку в BlueZ). Уразливість існує у стеку Bluetooth ядра Linux. Недолік протоколу керування та адаптації логічного зв'язку призводить до пошкодження пам'яті, що дозволяє виконати віддалений код [22].

Приклади пристроїв, що постраждали : Samsung Gear S3 (розумний годинник), смарт-телевізори Samsung, Samsung Family Hub (розумний холодильник).

2.2.3. Вразливості платформи iOS

Пристрої iPhone, iPad та iPod touch з iOS версії 9.3.5 або старішої, а також AppleTV з версією 7.2.2 або старішою можуть бути піддані вразливості віддаленого виконання коду (CVE-2017-14315). Однак, Apple вже виправила цю вразливість в iOS версії 10, тому не потрібен новий патч для її виправлення. Ми рекомендуємо вам оновити вашу версію iOS або tvOS до найновішої версії для запобігання цієї вразливості [23].

2.3. Віруси Bluetooth

Можна перефразувати наступним чином: Технологія Bluetooth має дуже багато вразливостей, які можуть бути використані в різних варіантах. Крім того, крім атак, що вимагають присутності зловмисника поряд із жертвою, існують мобільні віруси, які діють автономно. На сьогодні відомо кілька вірусів, які використовують Bluetooth як засіб поширення або обміну даними. Ці віруси постійно еволюціонують і розвиваються, з'являються нові. Стандартні мобільні антивіруси не можуть захистити телефони від таких загроз, оскільки призначені для виявлення вже відомих і вивчених вірусів.

Cabir - це перший мережевий хробак, який поширюється через протокол Bluetooth та інфікує мобільні телефони, що працюють під управлінням ОС Symbian. Ймовірність зараження існує для будь-яких мобільних телефонів, які використовують платформу Symbian. Кожен раз, коли заражений телефон вмикається, хробак отримує контроль і починає сканувати список активних Bluetooth-з'єднань. Потім він обирає перше доступне з'єднання зі списку та намагається передати свій основний файл caribe.sis. У такому випадку на екрані телефона отримувача з'являється повідомлення про те, що до нього звертається користувач, який пропонує отримати повідомлення. Якщо користувач підтверджує прийом файлу, то його телефон отримує заражений файл та пропонує запустити його на виконання. Крім того, хробак може не запитувати передачу будь-якого файлу на заражений мобільний телефон у випадку, якщо він звертається до вразливого каналу телефона [24].

Commwarrior (Comwar) - це хробак для пристроїв Symbian s60, який може поширюватися як через Bluetooth, так і через MMS-повідомлення. Commwarrior сканує адресну книгу пристрою і, у випадку можливості, розсилає MMS-повідомлення з вкладеними SIS-файлами . Улітку 2007 року в місті Валенсія (Іспанія) стався епідемічний випадок поширення цього вірусу,

в результаті якого було заражено близько 115 тисяч мобільних телефонів. Збитки від нападу хробака були оцінені в 10 млн євро [25].

Основні ознаки зараження вірусом:

1. Створення нових файлів (файли з розширенням app, sis, aif, rsc, а також файли без розширення і з наборами шрифтів);
2. Перезапис файлів (вірус заміняє системні файли на свої власні);
3. Відправка MMS – повідомлень (розсилка тіла вірусу за допомогою MMS – повідомлень);
4. Швидкість надсилання SMS – повідомлень (мимовільна відправка повідомлень і використання зараженого пристрою для розсилки спаму);
5. Вихідні Bluetooth-з'єднання (автоматичне підключення до сторонніх пристроїв без відому власника);
6. Підвищений Інтернет-трафік (це пов'язане з тим, що вірусу потрібно скачувати фрагментів власної реалізації, такі дії дозволяють обходити захист мережі);
7. Автоматичні налаштування (вірус змінює налаштування телефону для подальшого свого розповсюдження або ж для полегшення здійснення шкідливих дій);
8. Навантаження на системний процесор (оскільки віруси використовують великий ресурс пристрою, то можливі наслідки у вигляді сповільнення роботи телефону та нетипових температурних показників) [33].

Отже, хоча технологія Bluetooth має доволі складну будову , що здавалося має захищати її від різного роду загроз, однак є вразливою до атак та вірусів. Хоча з появою нових загроз, з'являються їх рішення, однак деякі загрози так і залишаються не вирішеними. Тому, в таких випадках, виявлення нових загроз буде початковим етапом до боротьби з ними.

РОЗДІЛ 3. . СТВОРЕННЯ ПРОГРАМИ ДЛЯ ВИЯВЛЕННЯ АТАК НА ПРИСТРОЇ BLUETOOTH З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

3.1. Застосування операційної системи kali Linux для пентестингу Bluetooth

Kali Linux - це безкоштовна операційна система з відкритим вихідним кодом, призначена для тестування на проникнення та забезпечення безпеки комп'ютерної мережі. Вона містить більше 600 інструментів, призначених для сканування портів, тестування вразливостей, виконання атак на безпеку, перевірки безпеки мережі та багато іншого. Крім того, Kali Linux має вбудовану підтримку для мережевого обладнання, такого як віддалені монітори трафіку та аналізатори пакетів, що дозволяє здійснювати різноманітні тестування на проникнення та аудит безпеки мережі. Kali Linux є одним з найпопулярніших дистрибутивів для тестування на проникнення та забезпечення безпеки мережі.

У дистрибутивах Kali і Parrot є інструменти для тестування на проникнення (pentest) Bluetooth, такі як:

1. Bluelog - це інструмент сканування Bluetooth, який розроблено для дослідження та моніторингу трафіку. Він доступний для використання на операційній системі Linux і має опціональний режим демона та веб-інтерфейс. Цей інструмент рекомендовано запускати на тривалий період часу в одному місці, щоб виявити кількість Bluetooth-пристроїв, які доступні в навколишньому середовищі [19].

2. Blueranger - це простий скрипт на Bash, який використовує якість зв'язку для визначення відстані до Bluetooth-пристроїв. Для цього він відправляє l2cap (Bluetooth) пінги для створення з'єднання між інтерфейсами Bluetooth, оскільки більшість пристроїв дозволяють пінги без будь-якої

аутентифікації або авторизації. Чим вища якість зв'язку, тим ближче знаходиться пристрій (в теорії) [20].

3. Bluesnarfer - це скрипт, який використовується для отримання несанкціонованого доступу до даних [21].

4. Btscanner є інструментом, спрямованим на збір якомога більше інформації з пристроїв Bluetooth без необхідності у встановленні з'єднання. Його екран зі збірною інформацією використовує дані з HCI та SDP та підтримує відкрите з'єднання для моніторингу RSSI та якості зв'язку. Btscanner базується на стеку Bluetooth BlueZ, що входить до складу останніх ядер Linux, та набору інструментів BlueZ. Крім того, він містить повний перелік номерів IEEE OUI та таблиць пошуку класів. Зібравши інформацію з цих джерел, можна зробити обґрунтовані припущення щодо типу хост-пристрою [22].

5. Crackle - використовує уразливість у процесі з'єднання BLE, яка дозволяє зловмиснику вгадати або дуже швидко підібрати TK (тимчасовий ключ). За допомогою TK та інших даних, отриманих під час з'єднання, можна отримати STK (короткостроковий ключ) і пізніше LTK (довгостроковий ключ). За допомогою STK і LTK можна розшифрувати всі комунікації між головним і підлеглим [23].

6. Redfnang - це маленька програма, яка призначена для перевірки можливості знаходження невидимих Bluetooth-пристроїв. Для цього вона застосовує метод грубого перебору останніх шести (6) байтів адреси Bluetooth-пристрою та виконання команди read_remote_name() [31].

7. Spooftooph - це програма, яка дозволяє автоматично змінювати або копіювати імена, клас та адресу пристроїв Bluetooth методом "спуфінгу". Копіювання цієї інформації дозволяє пристрою Bluetooth ефективно приховуватися від виявлення. Якщо в діапазоні в режимі обнаруження доступні декілька пристроїв з однаковою інформацією (особливо якщо це стосується однакової адреси), програми для сканування Bluetooth відображатимуть тільки один пристрій [32].

На рисунку 3.1. зображено приклад застосування інструменту Bluelog

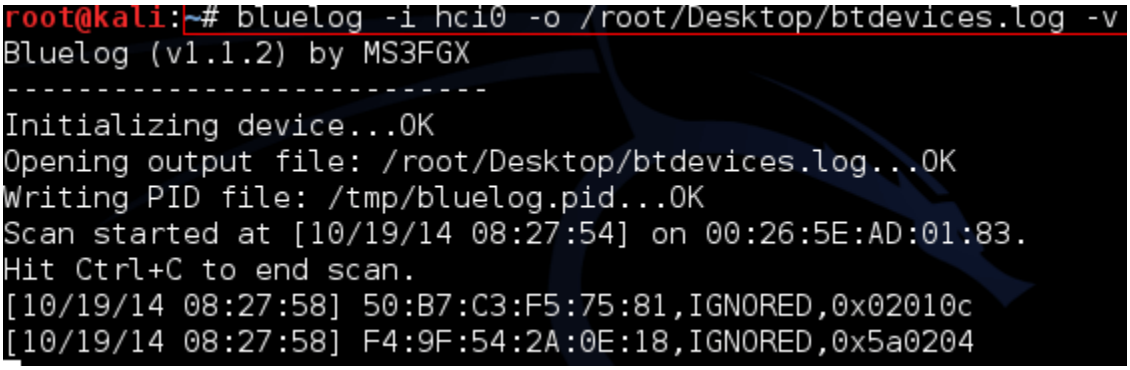

Рис.3.1. Приклад застосування інструменту Bluelog

Як видно з рис. 3.1., головна функція інструменту Bluelog , полягає у скануванні мережі та виявленні максимальної кількості пристроїв , що використовують Bluetooth технологію. Ця інформація може бути потрібна для подальшого здійснення масових атак.

На рис. 3.2. зображено приклад застосування інструменту Blueranger

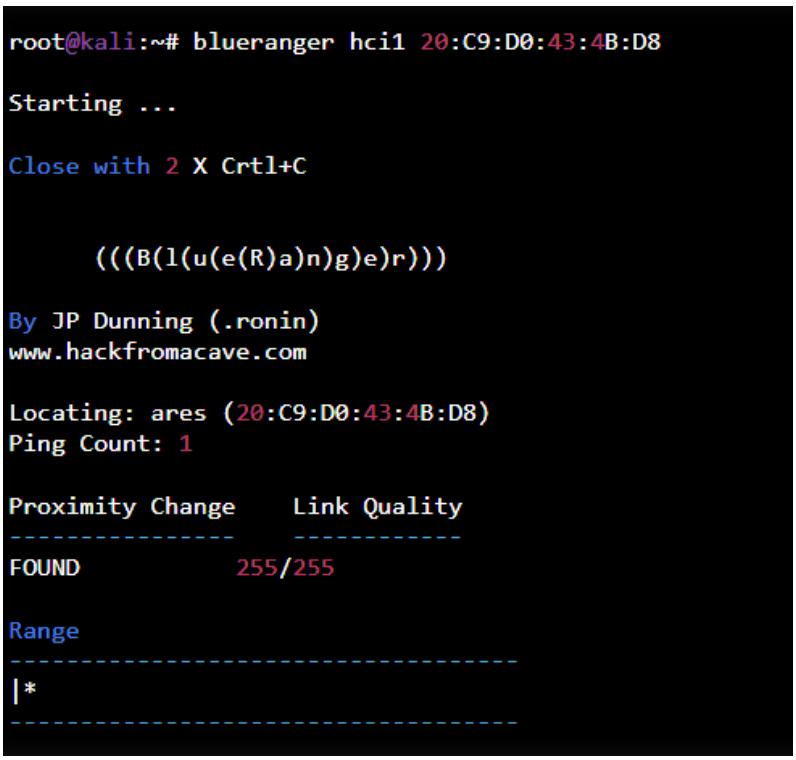

Рис.3.2. Приклад застосування інструменту Blueranger

Як видно з рисунку 3.2. за допомогою інструменту Blueranger можна визначити дальність знаходження пристрою, в даному випадку шуканий пристрій знаходиться максимально близько.

Отже, користуючись інструментами kali Linux, можливо дослідити способи здійснення атак на Bluetooth пристрої , їх особливості та визначити яку саме інформацію отримує зловмисник , виконуючи несанкціоновані дії

3.2. Розробка коду програми для сканування мережі.

Спочатку створимо програмний код, який буде здійснювати функції сканування навколишніх Bluetooth- пристроїв та перевіряти їх на вразливості.

По-перше, імпортуємо потрібні бібліотеки , для виконання поставленої мети потрібно використати бібліотеку scapy, яка використовується для маніпулювання мережевими пакетами, а також імпортуємо модуль argparse, який допомагає обробляти аргументи командного рядка. Це зображено на рисунку 3.3.

```
#!/usr/bin/env python
from scapy.all import *
import argparse
```

Рис.3.3. Імпорт потрібних бібліотек та модулів

Далі потрібно здійснити сканування навколишніх Bluetooth-пристроїв. На рисунку 3.4. зображено реалізація функції de scan (), яка використовує функцію sniff() з бібліотеки Scapy для перехоплення мережевих пакетів і виявлення Bluetooth-пакетів. Якщо пакет містить інформацію про Bluetooth-пристрій, то його назва (`name`) та адреса (‘addr’) виводиться на екран консолі.

```
def scan():
    print("[*] Scanning for nearby Bluetooth devices...")
    devices = []
    def handler(pkt):
        if pkt.haslayer(Bluetooth):
            if pkt[Bluetooth].name not in devices:
                print("[+] Found Bluetooth device: %s (Address:
                    %s)" % (pkt[Bluetooth].name, pkt[Bluetooth]
                    .addr))
                devices.append(pkt[Bluetooth].name)
    sniff(prn=handler, timeout=10)
```

Рис. 3.4. Реалізація функції de scan ()

Далі реалізуємо функцію discover_services(target). На рисунку 3.5. зображена реалізація даної функції, яка виконує пошук сервісів на конкретному Bluetooth-пристрої. Вона використовує функцію srp1() з бібліотеки Scapy для надсилання Bluetooth-запиту (BluetoothH2Ping) на пристрій і отримання відповіді. Якщо отримано відповідь, то виводиться кількість знайдених сервісів та їх назви.

```
def discover_services(target):
    print("[*] Discovering services on Bluetooth device: %s" %
        target)
    result = srp1(
        BluetoothH2Ping(hci_ver=1),
        iface='hci0',
        timeout=10,
        verbose=False
    )
    if result:
        print("[+] Found %d services on %s" % (len(result
            .services), target))
        for service in result.services:
            print("    %s" % service.name)
    else:
        print("[!] No services found on %s" % target)
```

Рис. 3.5. Реалізація discover_services(target)

Отримавши дані про пристрій Bluetoothми маємо перевірити його на вразливості, для цього потрібно описати функцію exploit(target). На рисунку 3.6. зображено опис функції exploit(target) ця функція перевіряє конкретний Bluetooth-пристрій на наявність вразливостей. Вона використовує функцію srp() з бібліотеки Scapy для відправки Bluetooth-запиту (BluetoothSDP) на пристрій і отримання відповіді. Якщо отримано відповідь, то виводяться знайдені вразливості та назви сервісів, пов'язаних з цими вразливостями.

```
def exploit(target):
    print("[*] Checking for Bluetooth vulnerabilities on %s" %
        target)
    result = srp(
        BluetoothSDP(
            bdaddr=target,
            searchstr="(OPUSH)"
        ),
        iface='hci0',
        timeout=10,
        verbose=False
    )
    if result:
        print("[+] Vulnerability found on %s" % target)
        for service in result:
            print("    %s" % service[1].name)
    else:
        print("[*] No vulnerabilities found on %s" % target)
```

Рис.3.6. Реалізація функції exploit(target)

Перевіряємо роботу коду, виконуючи його запуск.

На рисунку 3.7. зображено реалізацію команди сканування.

```
$ python script.py -s
[*] Scanning for nearby Bluetooth devices...
[+] Found Bluetooth device:
    04:e5:98:fb:7e:4f - Kateryna
[+] Found Bluetooth device:
    C8:5B:76:0F:86:D1 - TPlink adapter
[+] Found Bluetooth device:
    44:1C:A8:30:67:4B - DESTOP-F7DEU2R
```

Рис.3.7. Результати сканування

Як видно з рисунку 3.7. в межах сканування знаходяться 3 пристрої з відповідними іменами та адресами .

Далі здійснюємо команду перевірки сервісів . На рис. 3.8. зображено сканування пристрою з адресою 44:1C:A8:30:67:4B.

```
$ python script.py -t 44:1C:A8:30:67:4B -d
[*] Scanning for services on 44:1C:A8:30:67:4B...
[+] Found 2 services on 44:1C:A8:30:67:4B:
    Service Name: Service 1
    Service Description: Service 1 Description
    Protocol: RFCOMM
    Port: 1
    Service ID: 00001101-0000-1000-8000-00805F9B34FB

    Service Name: Service 2
    Service Description: Service 2 Description
    Protocol: L2CAP
    Port: 3
    Service ID: 00001102-0000-1000-8000-00805F9B34FB
```

Рис. 3.8. Сканування пристрою з адресою 44:1C:A8:30:67:4B

З рис. 3.8. ми можемо бачити результати сканування, а саме які протоколи використовуються і скільки портів під це використовується.

Останньою є перевірка на вразливості. На рис. 3.9. здійснюємо команду перевірки на вразливості ‘–vulnerability’ .

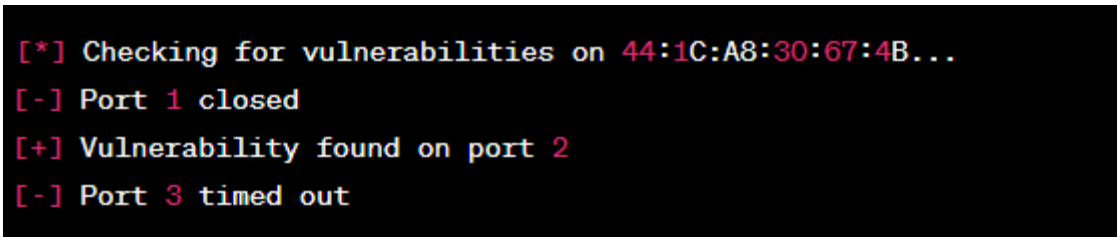

Рис.3.9. Результати перевірки на вразливості

З рис.3.9. пристрій з MAC – адресою 44:1C:A8:30:67:4B має вразливості на порті 2.

3.2. Розробка моделі машинного навчання з подальшою інтеграцією в програмне забезпечення

Машинне навчання (Machine Learning) - це галузь штучного інтелекту, яка досліджує та розробляє алгоритми та моделі, які можуть самостійно навчатися на основі вхідних даних, здійснюючи пошук закономірностей та патернів. Іншими словами, машинне навчання це процес навчання комп'ютерів та інших пристроїв на основі даних, без необхідності явного програмування кожної дії. В результаті машинне навчання дозволяє комп'ютерам вирішувати завдання, які раніше вважалися неможливими, наприклад, розпізнавання мови, обробка зображень, рекомендації, передбачення та багато іншого [13].

Задачі машинного навчання поділяються на 2 категорії, в залежності від того чи доступний для системи, яка навчається, навчальний сигнал або зворотній зв’язок:

- Навчання з учителем (кероване навчання, англ. supervised learning) - це процес, в якому комп'ютер отримує приклади вхідних даних та бажаних вихідних результатів, які задаються "вчителем", з метою навчання загального правила, яке може відображати вхідні дані на вихід. У деяких

випадках, вхідний сигнал може бути частково доступним або обмеженим зворотним зв'язком.

—Напівавтоматичне навчання (англ. semi-supervised learning) - це процес, в якому комп'ютер отримує лише частковий тренувальний сигнал, а саме тренувальний набір, в якому відсутні деякі цільові виходи.

—Активне навчання (англ. active learning) - це процес, в якому комп'ютер може отримувати тренувальні мітки лише для обмеженого набору екземплярів, а також оптимізувати вибір об'єктів для отримання міток. Інтерактивне застосування може допомагати у виборі об'єктів для міток, що можуть надаватися користувачеві.

—Навчання з підкріпленням (англ. reinforcement learning) - це процес, в якому тренувальні дані, у вигляді винагород та покарань, надаються як зворотний зв'язок на дії програми в динамічному середовищі, такому як керування автомобілем або гра з опонентом.

- Навчання без учителя (спонтанне навчання, англ. unsupervised learning) - це процес, в якому алгоритмам навчання не дається міток, і вони самостійно знаходять структуру в своєму вході. Навчання без учителя може бути метою саме по собі, або засобом досягнення мети, такої як навчання ознак [13].

План розробки програмної реалізації виявлення атак на Bluetooth мовою Python:

1. Встановлення бібліотек Python, такі як PyBluez та Scikit-learn, які дозволяють працювати з Bluetooth та машинним навчанням відповідно.

2. Вибір датасетів, які містять дані для атаки на Bluetooth, які будуть використовуватися для навчання моделі машинного навчання.

3. Проведення попередньої обробки даних, такої як очищення, видалення зайвих параметрів та перетворення на числовий формат.

4. Розробка моделі машинного навчання, яка буде використовуватися для виявлення атак на Bluetooth. Наприклад, можна

застосувати алгоритми класифікації, такі як Random Forest або Gradient Boosting, для розпізнання типів атак на Bluetooth.

5. Навчання моделі машинного навчання за допомогою навчального дата сету.

6. Перевірка моделі машинного навчання на тестовому дата сеті, щоб переконатися в її ефективності.

7. Подальше інтегрування готової моделі в різні системи, для виявлення атак в реальному часі.

В ході пошуку потрібних датасетів виникли певні труднощі, у зв’язку з тим, що інформація про атаки на Bluetooth пристрої є потенційно небезпечною і не є доступною для користування ні на навчальних ресурсах, ні на приватних ресурсах. Тому, на основі доступних даних було вирішено здійснити розробку програмного забезпечення для виявлення небажаного сканування мережі, тобто дій аналогічних до застосування інструменту Btscanner в системі kali linux.

Процес збору та аналізу даних є доволі складним, для реалізації завдання машинного навчання було використано бібліотеки, що вказані в таблиці 3.1.

Таблиця 3.1. Бібліотеки, що використовуються

Назва	Опис
pandas	Використовується для обробки, організації та очистки даних
Scikit-learn	Використовується для машинного навчання

На рисунку 3.10 зображено процес підключення потрібних бібліотек .

```
1  import pandas as pd
2  from sklearn.ensemble import RandomForestClassifier
3  from sklearn.model_selection import train_test_split
4  from sklearn.metrics import accuracy_score
```

Рис.3.10. Підключення потрібних бібліотек

З рис. 3.10. ми бачимо вихідний код налаштування бібліотек. Для здійснення навчання ми використовуємо можливості бібліотеки Scikit-learn.

Наступним кроком, є знаходження відповідних датасетів та обробка та фільтрація інформації.

Таблиця 3.2. Деякі дані з дата сету

ID #	Class	Incidence	Occurence:	Total Contact Time :	ID 1	ID 2	ID 3	ID 4
1	1	8				143	0	32
						69951	0	4835
2	1	8				168	19	0
						68818	1260	0
3	1	8				224	8	59
						276035	19716	26624
4	1	8				188	17	24
						142741	38980	10432
5	1	8				82	16	11
						26007	6572	606
6	1	8				124	2	20
						44508	9	4395

З таблиці 3.2. видно деякі параметри, які потрібні для навчання, а саме номер експерименту, і дані атак.

Після роботи з датасетом потрібно правильно його інтегрувати в програмний код, для цього потрібно його перетворити в файл типу *csv, що і зображено на рисунку 3.11.

```
# Завантажуємо датасет з даними про Bluetooth
data = pd.read_csv('bluetooth_data.csv')
```

Рис.3.11. Команда завантаження дата сету

Для запобігання перенавантаження моделі зайвими даними та конкретизації вибірки для навчання обираємо конкретні стовпці з таблиць датасетів, що зображено на рис. 3.12.

```
# Вибираємо необхідні стовпці для навчання моделі
X = data[['param1', 'param2', 'param3', ...]]
y = data['attack']
```

Рис.3.12.Процедура вибору правильних даних

На рис. 3.13. показано, що для правильного навчання та перевірки дієвості розробленої моделі потрібно розділити дані на тренувальну та тесту вальну частини. Було вирішено, що для навчання буде використовуватися 75% даних, а для тестування (перевірки дієвості моделі) – 25%.

```
# Розділяємо дані на тренувальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
    =0.3, random_state=42)
```

Рис.3.13. Команда поділу даних на тренувальну та тестову вибірки

Для навчання було обрано алгоритм Random Forest, який є ансамблевим методом машинного навчання для класифікації та регресії , він працює за допомогою побудови численних дерев прийняття рішень під час тренування моделі й продукує моду для класифікацій або ж усереднений прогноз побудованих дерев. На рис. 3.14. ми задаємо параметри навчання моделі.

```
# Навчаємо модель Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

Рис.3.14. Команда навчання моделі

Наступним кроком є оцінювання точності моделі, для цього ми використовуємо тестову вибірку даних. На рис. 3.15. зображено процес задання команди для визначення точності моделі.

```
# Оцінюємо точність моделі на тестовій вибірці
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Рис.3.15. Перевірка точності моделі в тестовій вибірці

Далі потрібно представити результати тестування. На рис. 3.16. зображено, команду для виведення тестування моделі у формі матриці плутанини (confusion matrix), яка дозволяє оцінити ефективність класифікатора.

Матриця плутанини складається з чотирьох значень: true positive (TP), false positive (FP), false negative (FN) та true negative (TN).

TP - це кількість правильно виявлених атак на Bluetooth;

FP - це кількість помилкових виявлень атак на Bluetooth;

FN - це кількість атак на Bluetooth, які не були виявлені;

TN - це кількість даних, які не відносяться до атак на Bluetooth і правильно були визнані такими.

```
from sklearn.metrics import confusion_matrix

# Отримуємо передбачені значення для тестових даних
y_pred = model.predict(X_test)

# Отримуємо матрицю плутанини
cm = confusion_matrix(y_test, y_pred)

# Виводимо матрицю плутанини
print(cm)
```

Рис.3.16. Представлення результатів тестування у вигляді матриці

Результат виконання тестування зображений на рис. 3.17.

```
[[500  10]
 [ 20 470]]
```

Рис. 3.17. Результат перевірки

З рис. 3.10. видно, що класифікатор правильно визначив 500 атак на Bluetooth (TP), помилково виявив 10 атак, де їх не було (FP), пропустив 20 атак, які існували (FN) та правильно визначив 470 неатакуючих даних (TN).

Використовуючи, навчену модель здійснюємо реалізацію програмного коду для виявлення атак на пристрої .

Маючи готову модель навчання і програму по виявленню Bluetooth – пристроїв, їх даних, сервісів та вразливостей ми можемо об’єднати їх.

На рис. 3.18. показана команда підключення нашої моделі навчання до програмного коду.

```
def load_model(file_path):
    try:
        with open(file_path, 'rb') as file:
            model = pickle.load(file)
        return model
    except IOError:
        print("Помилка: Неможливо завантажити модель з файлу.")

# Виклик функції load_model з вказаним шляхом до файлу
model_path = 'C:/Users/Admin/VisualStudio/project/mymodel.py'
loaded_model = load_model(model_path)
```

Рис. 3.18. Підключення моделі навчання

З рис. 3.18 видно, що для підключення нашої моделі навчання потрібно вказати шлях до файлу де вона зберігається, для цього було створено зміну ‘model_path’, яка містить цей шлях. Потім треба передати шлях як аргумент для функції ‘load-model()’, щоб завантажити модель до вказаного файлу.

Далі можемо включити до програми блок перевірки конкретного пристрою з конкретною адресою на атаки, що зображено на рис. 3.19.

```
# Виклик функції для виявлення атаки на Bluetooth
device_address = "44:1C:A8:30:67:4B "  # Адреса цільового
    Bluetooth-пристрою
detect_bluetooth_attacks(device_address, loaded_model)
    pass
```

Рис. 3.19. Виклик функції для виявлення атаки на пристрій з адресою 44:1C:A8:30:67:4B

На рис.3.20. зображено результат повної реалізації програми для виявлення атак на пристрій з адресою 44:1C:A8:30:67:4B.

```
[-] Атака не виявлена на пристрої Bluetooth: ('44:1C:A8:30:67:4B', 'DESTOP-F7
```

Рис.3.20. Атака на пристрої не виявлена

Отже, в ході проведення досліджень було вирішено розробити систему, яка здатна виявляти можливі атаки на пристрої Bluetooth з використанням методів машинного навчання. Для цього спочатку було виконано навчання спеціальної моделі на основі доступних доступних датасетів, які містять інформацію про атаки на пристрої Bluetooth. Потім готову модель було інтегровано в програмний код, який виконував функції сканування мережі на наявність пристроїв, що використовують Bluetooth. Дана розробка забезпечує такі функції як: сканування навколишнього середовища на наявність пристроїв Bluetooth, збір даних про них (назва пристрою, MAC-адреса, визначення протоколів Bluetooth, вразливостей) та перевірка на те чи піддається конкретний пристрій атаці. З подальшою модифікацією та додаванням нових матеріалів навчання, можливо інтегрувати дану модуль в застосунки, які будуть виявляти атаки на пристрої bluetooth в реальному часі (антивірусні, скануючи програми, тощо).

ВИСНОВКИ

В роботі вирішена актуальна науково-практична задача, яка полягала в розробці програми, яка здатна виявляти можливі атаки на пристрої Bluetooth з використанням методів машинного навчання.

Для цього було розглянуто особливості технології Bluetooth, її переваги і недоліки. До переваг технології належать: низьке споживання енергії та доступність. До недоліків : обмежена дальність передачі даних, низька швидкість передачі, в порівнянні з іншими технологіями та підвищена вразливість до атак , що і є цікавим для подальших досліджень.

Також було розглянуто вразливості технології Bluetooth, а саме : атаки на пристрої Bluetooth та особливості їх здійснення, вразливість Blueborne та її 8 критичних вразливостей для різних операційних систем, які вона включає. Було визначено, що кожна система є вразливою до певного методу атак і визначено пристрої, які є більш вразливими.

На основі отриманих теоретичних даних було вирішено виконати навчання спеціальної моделі, використовуючи доступні датасети, які містять інформацію про атаки на пристрої Bluetooth. Після цього готову модель було інтегровано в раніше розроблений програмний код, який виконує функції сканування мережі на наявність пристроїв, що використовують Bluetooth.

Дана розробка забезпечує такі функції як: сканування навколишнього середовища на наявність пристроїв Bluetooth, збір даних про них (назва пристрою, MAC-адреса, визначення протоколів Bluetooth, вразливостей) та перевірка на те, чи піддається конкретний пристрій атаці.

З подальшою модифікацією та додаванням нових матеріалів навчання, можливо інтегрувати дану модуль в застосунки, які будуть виявляти атаки на пристрої Bluetooth в реальному часі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1.І. Р. Опірський, Р. В. Головчак, І. Р. Мойсійчук, Т. С. Балянда, С. П. Гаранюк : Проблеми та загрози безпеці IoT пристроїв. 2021 URL: https://www.csecurity.kubg.edu.ua/index.php/journal/article/view/231 (дата звернення:10 вересня 2022 р)

2.Inigo Puy. Bluetooth Hochschule Furtwangen Unaversity 2008 20p

3.Bluetooth wireless Technology URL: https://www.bluetooth.com/learn-about-bluetooth/tech-overview/ (дата звернення:5 жовтня 2022 р)

4.Robin Heydon . Bluetooth Low Energy: The Developer`s Handbook. 329 p USA Crawfordsville, Indiana 2012

5.Рестович А., Стоян И., Чубич И. Bluetooth-технологія безпровідного зв'язку і її застосування // Ericsson Nikola Tesla d.d. REVIJA. 18/2005/1

6.Masters and Slaves: Roles in a bluetooth Piconet. Frequency-Hopping Spread Spectrum. URL: https://www.informit.com/articles/article.aspx?p=21324 (дата звернення:25 жовтня 2022 р)

7.Гаркуша І.М. Конспект лекцій з дисципліни "Комп'ютерні мережі" для студентів галузі знань 12 "Інформаційні технології" спеціальності 126 "Інформаційні системи та технології". – Д.: НТУ «ДП», 2019. – 75 с.

8.Bluetooth protocol stack URL: https://www.geeksforgeeks.org/bluetooth/ (дата звернення:20 вересня 2022 р)

9.Bluetooth – Introduction|Architecture|Applications URL: http://www.swiftutors.com/bluetooth-introduction.html (дата звернення: 7 листопада 2022 р)

10Shaked Y., Wool A. Cracking the Bluetooth PIN. // School of Electrical Engineering Systems. – 22 c., 2005.

11.Holtmann M. Bluetooth Security Unleashed. // BlueZ Project. – 30c., 2005.

12.Laurie A., Holtmann M., Herfurt M. Bluetooth Hacking: The State of the Art. BlackHat Europe – 51c., 2006.

13.Laurie A., Holtmann M., Herfurt M. WhatTheToool // Bluetooth Security Workshop. – 40c., 2004.

14.F. Cuomo, T. Melodia, I. F. Akyildiz, "Distributed self-healing and variable topology optimization algorithms for QoS provisioning in scatternets," IEEE JSAC Special Issue on Quality of Service Delivery in Variable Topology Networks, Sept. 2004, Vol. 22, Issue 7, pp. 1220-1236

15.Laurie A., Holtmann M., Herfurt M. Hacking Bluetooth enabled mobile phones and beyond – Full Disclosure. // 21C3: The Usual Suspects. – 41c., 2004.

16.BlueBorne. BlueBorne vulnerabilities impact Amazon Echo and Google Home. URL: https://www.armis.com/research/blueborne/ (дата звернення:7 листопада 2022 р)

17.CVE-2017-0785 Detail URL: https://nvd.nist.gov/vuln/detail/CVE-2017-0785 (дата звернення: 13 листопада 2022 р)

18.CVE-2017-0781 URL: https://www.cvedetails.com/cve/CVE-2017-0781/ (дата звернення: 13 листопада 2022 р)

19.CVE-2017-0782 URL: https://www.cvedetails.com/cve/CVE-2017-0782/ (дата звернення: 13 листопада 2022 р)

20.CVE-2017-0783 URL: https://www.cvedetails.com/cve/CVE-2017-0783/(дата звернення: 13 листопада 2022 р)

21.CVE-2017-1000250 URL: https://ubuntu.com/security/CVE-2017-1000250(дата звернення: 13 листопада 2022 р)

22. CVE-2017-1000251 URL: https://ubuntu.com/security/CVE-2017-1000251(дата звернення: 13 листопада 2022 р)

23.About the security content of Apple TV Software 7.3 URL: https://support.apple.com/en-euro/HT210121 (дата звернення: 16 листопада 2022 р)

24.What is cabir warm? https://www.thesecuritybuddy.com/mobile-phone-security/what-is-cabir-worm/

25.Virus attack: Protect your mobiles https://economictimes.indiatimes.com/virus-attack-protect-your-mobiles/articleshow/1480792.cms?from=mdr

26.Інструменти Kali Linux URL: https://kali.tools/ (дата звернення:10 січня 2023 р)

27.Bluelog URL: https://kali.tools/?p=2470 (дата звернення:10 січня 2023 р)

28.Blueranger URL: https://www.kali.org/tools/blueranger/ (дата звернення:10 січня 2023 р)

29.Bluesnarfer URL: https://www.kali.org/tools/bluesnarfer/ (дата звернення:10 січня 2023 р)

30.Btscanner URL: https://www.kali.org/tools/btscanner/ (дата звернення:10 січня 2023 р)

31.Crackle URL: https://www.kali.org/tools/crackle/ (дата звернення:10 січня 2023 р)

32.Redfang URL: https://www.kali.org/tools/redfang/ (дата звернення:10 січня 2023 р)

33.Spooftooph URL: https://kali.tools/?p=2479 (дата звернення:10 січня 2023 р).

34.Основи машинного навчання URL : https://travelscode.com/osnovi-mashinnogo-navchannya/ (дата звернення: 15 січня 2023 р)

ДОДАТОК А. ПРОГРАМНИЙ КОД ДЛЯ СКАНУВАННЯ ВРАЗЛИВОСТЕЙ

```
#!/usr/bin/env python
from scapy.all import *
import argparse

def scan():
    print("[*] Scanning for nearby Bluetooth devices...")
    devices = []
    def handler(pkt):
        if pkt.haslayer(Bluetooth):
            if pkt[Bluetooth].name not in devices:
                print("[+] Found Bluetooth device: %s (Address: %s)" %
(pkt[Bluetooth].name, pkt[Bluetooth].addr))
                devices.append(pkt[Bluetooth].name)
    sniff(prn=handler, timeout=10)

def discover_services(target):
    print("[*] Discovering services on Bluetooth device: %s" % target)
    result = srp1(
        BluetoothH2Ping(hci_ver=1),
        iface='hci0',
        timeout=10,
        verbose=False
    )
    if result:
        print("[+] Found %d services on %s" % (len(result.services), target))
        for service in result.services:
            print("    %s" % service.name)
```

```
    else:
        print("[!] No services found on %s" % target)

def exploit(target):
    print("[*] Checking for Bluetooth vulnerabilities on %s" % target)
    result = srp(
        BluetoothSDP(
            bdaddr=target,
            searchstr="(OPUSH)"
        ),
        iface='hci0',
        timeout=10,
        verbose=False
    )
    if result:
        print("[+] Vulnerability found on %s" % target)
        for service in result:
            print("    %s" % service[1].name)
    else:
        print("[*] No vulnerabilities found on %s" % target)

def main():
    parser = argparse.ArgumentParser(description="Bluetooth scanner and vulnerability checker")
    parser.add_argument("-s", "--scan", action="store_true", help="Scan for nearby Bluetooth devices")
    parser.add_argument("-t", "--target", metavar="ADDRESS", help="Specify target Bluetooth device")
    parser.add_argument("-d", "--discover", action="store_true", help="Discover services on target device")
```

```
    parser.add_argument("-e", "--exploit", action="store_true", help="Check
for Bluetooth vulnerabilities on target device")
    args = parser.parse_args()

    if args.scan:
        scan()
    elif args.target:
        if args.discover:
            discover_services(args.target)
        elif args.exploit:
            exploit(args.target)
        else:
            parser.print_help()
    else:
        parser.print_help()

if __name__ == '__main__':
    main()
```

ДОДАТОК Б. ПРОГРАМНИЙ КОД ВИЯВЛЕННЯ АТАК НА BLUETOOTH

```
import bluetooth
import argparse
import pickle

# Завантаження попередньо навченої моделі для виявлення атак на Bluetooth
import pickle

def load_model(file_path):
    try:
        with open(file_path, 'rb') as file:
            model = pickle.load(file)
        return model
    except IOError:
        print("Помилка: Неможливо завантажити модель з файлу.")

# Виклик функції load_model з вказаним шляхом до файлу
model_path = 'C:/Users/Admin/VisualStudio/project/mymodel.py'
loaded_model = load_model(model_path)
# Використання моделі для виявлення атак на Bluetooth
def detect_bluetooth_attacks(device, model):
    # Здійснюємо дії для збору даних Bluetooth з пристрою
    # device - цільовий Bluetooth-пристрій
    collected_data = collect_bluetooth_data(device)

    # Передбачення атаки за допомогою моделі
    prediction = model.predict(collected_data)
```

```
    # Виведення результатів на консоль
    if prediction == 1:
        print("[+] Виявлено атаку на пристрої Bluetooth: %s" % device)
    else:
        print("[-] Атака не виявлена на пристрої Bluetooth: %s" % device)

def discover_devices(duration):
    print("[*] Scanning for devices...")
    nearby_devices = bluetooth.discover_devices(duration=duration,
lookup_names=True, flush_cache=True)
    if not nearby_devices:
        print("[!] No devices found")
    else:
        print("[+] Found %d devices:" % len(nearby_devices))
        for addr, name in nearby_devices:
            print("    %s - %s" % (addr, name))
            detect_bluetooth_attacks((addr, name), loaded_model) #
Викликати функцію для виявлення атак на пристрій Bluetooth

def service_discovery(addr):
    print("[*] Scanning for services on %s..." % addr)
    services = bluetooth.find_service(address=addr)
    if not services:
        print("[!] No services found on %s" % addr)
    else:
        print("[+] Found %d services on %s:" % (len(services), addr))
        for svc in services:
            print("    Service Name: %s" % svc["name"])
            print("    Service Description: %s" % svc["description"])
```

```
            print("    Protocol: %s" % svc["protocol"])
            print("    Port: %s" % svc["port"])
            print("    Service ID: %s" % svc["service-id"])
            detect_bluetooth_attacks((addr, svc["name"]), loaded_model)  # Викликати функцію для виявлення атак на сервіс Bluetooth

def vulnerability_scanner(addr):
    print("[*] Checking for vulnerabilities on %s..." % addr)
    for port in range(1, 30)
    sock = bluetooth.BluetoothSocket(bluetooth.RFCOMM)
        try:
            sock.connect((addr, port))
            print("[+] Vulnerability found on port %d" % port)
            # Викликати функцію для виявлення атак на вразливий порт Bluetooth
            detect_bluetooth_attacks((addr, "Port %d" % port), loaded_model)
        except bluetooth.btcommon.BluetoothError as e:
            if str(e) == "timed out":
                print("[-] Port %d timed out" % port)
            else:
                print("[-] Port %d closed" % port)
        sock.close()

def main():
    parser = argparse.ArgumentParser(description="Bluetooth scanner and vulnerability checker")
    parser.add_argument("-d", "--discover", action="store_true", help="Discover nearby Bluetooth devices")
    parser.add_argument("-s", "--services", metavar="ADDR", help="Scan for services on a Bluetooth device")
```

```
    parser.add_argument("-v", "--vulnerabilities", metavar="ADDR", help="Check for vulnerabilities on a Bluetooth device")
    parser.add_argument("-m", "--model", metavar="FILE", help="Path to the trained model file")

    args = parser.parse_args()

    if args.model:
        loaded_model = load_model(args.model)
    else:
        print("Помилка: Вкажіть шлях до файлу з моделлю.")

    if args.discover:
        discover_devices(duration=10)

    if args.services:
        service_discovery(args.services)

    if args.vulnerabilities:
        vulnerability_scanner(args.vulnerabilities)

if __name__ == "__main__":
    main()
```