VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

DETEKCE BEZPEČNOSTNÍCH INCIDENTŮ V BLUETOOTH SÍTÍCH

DETECTION OF SECURITY INCIDENTS IN BLUETOOTH NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **BRONISLAV BÁRTEČEK**
AUTHOR

VEDOUCÍ PRÁCE **Ing. JAN KOŘENEK, Ph.D.**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačových systémů (UPSY) Akademický rok 2018/2019

Zadání bakalářské práce

Student: **Bárteček Bronislav**
Program: Informační technologie
Název: **Detekce bezpečnostních incidentů v Bluetooth sítích**
Detection of Security Incidents in Bluetooth Networks
Kategorie: Bezpečnost

Zadání:

1. Seznamte se s komunikačním protokolem Bluetooth Low Energy (BLE) a možnostmi monitorovacího nástroje Ubertooth.
2. Vytvořte testovací prostředí pro analýzu BLE komunikace s cílem zachytit vzory provozu vybraného zařízení a odeslat zachycené vzorky provozu.
3. Navrhněte a implementujte vhodné způsoby detekce bezpečnostních problémů v BLE zařízeních.
4. Vytvořenou implementaci ověřte v připraveném testovacím prostředí.
5. V závěru práce diskutujte dosažené výsledky.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Kořenek Jan, Ing., Ph.D.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 26. října 2018

Abstrakt

Cieľom tejto bakalárskej práce je detekovať bezpečnostné incidenty v Bluetooth Low Energy (BLE) sieťach. Bolo potrebné vytvoriť nástroj, ktorý by odhalil bezpečnostné problémy v zariadeniach a monitoroval aktivitu zariadení komunikujúcich pomocou protokolu BLE. Pri riešení bol použitý nástroj Ubertooth umožňujúci sledovanie BLE komunikácie. Nástroj Ubertooth je použitý na zachytenie BLE paketov. Tie následne vytvorený program dekóduje a vykoná ich analýzu. Vytvorený nástroj určí z týchto dát mieru zabezpečenia zariadení. Zároveň monitoruje aktivitu siete a prípadne informuje používateľa o nechcených aktivitách zariadení, ako je napríklad pripájanie cudzích zariadení na zariadenia používateľa.

Abstract

The goal of this bachelor thesis is to detect security incidents in Bluetooth Low Energy(BLE) networks. It was necessary to create a tool that would detect security issues in devices and monitor the activity of devices that communicate using BLE. Ubertooth was used in the solution to sniff BLE communication. Ubertooth is used to capture BLE packets, which are then decoded by the created program and analyzed. The created tool determines the device security rate from these data. At the same time, it monitors network activity and, if necessary, informs the user of unwanted device activities, such as connecting foreign devices to user devices.

Kľúčové slová

Bluetooth Low Energy, BLE, Ubertooth, BLE komunikácia, detekcia problémov BLE, detekcia, bezpečnostné incidenty BLE, BLE zariadenia.

Keywords

Bluetooth Low Energy, BLE, Ubertooth, BLE communication, detection of incidents in BLE, detection, Security Incidents in BLE, BLE devices.

Citácia

BÁRTEČEK, Bronislav. *Detekce bezpečnostních incidentů v Bluetooth sítích.* Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Kořenek, Ph.D.

Detekce bezpečnostních incidentů v Bluetooth sítích

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Jana Kořenka, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Bronislav Bárteček
12. mája 2019

Poďakovanie

Rád by som sa poďakoval hlavne vedúcemu mojej bakalárskej práce pánovi Ing. Jánovi Kořenkovi, Ph.D., za odborné vedenie pri tvorbe tejto bakalárskej práce. Takisto veľké ďakujem patrí Ing. Radkovi Krejčímu za pomoc a rady.

Obsah

Kapitola 1

Úvod

Žijeme v rýchlo sa vyvýjajúcej dobe. Sme obklopení modernými technológiami, ktoré nás ovplyvňujú, či už chceme alebo nie. Robia nám spoločníkov v každodennom živote. Kávovar, chladnička alebo hodinky, všetko sa stáva v dnešnej dobe „smart“. Teda inteligentnými zariadeniami, ktoré je často možné ovládať pomocou našich „smartphonov“. „Smartphone“, teda minipočítač s možnosťou telefonovania, ktorý nahradil klasické telefóny, vlastní skoro každý z nás. A aj preto niet divu, že práve toto zariadenie sa stalo tým, pomocou ktorého vieme ovládať iné inteligentné zariadenia. Samozrejme, že káble nahradila v dnešnej dobe bezdrôtová komunikácia. Jedným z bezdrôtových protokolov, ktorým vie komunikovať takmer každý smartphone, je Bluetooth. Pre inteligentné zariadenia a ich ovládanie sa požíva konkrétne Bluetooth Low Energy. Vďaka nemu vieme dotykom prsta na obrazovke smartphonu zmeniť teplotu na termostate, zasvietiť svetlo či odomknúť dvere.

Inteligentné zariadenia si môže každý z nás voľne kúpiť. Ak sa rozhodneme pre inteligentné zariadenie s komunikačným protokolom Bluetooth, bude určite používať Bluetooth Low Energy(BLE). BLE je súčasťou Bluetooth od verzie 4.0[4]. Jedným z dôvodov použitia tejto verzie Bluetooth je, že inteligentné zariadenia sú často napájané z batérií. BLE zariadenia majú často malé rozmery a tým pádom aj malé kapacity batérií. Táto verzia Bluetooth je zameraná na vyššiu výdrž batérie a funguje rozdielne oproti klasickej verzii Bluetooth. BLE sa líši v spôsobe párovania, odosielania dát a šifrovania[3]. Aj napriek tomu, že tieto zariadenia odosielajú často citlivé informácie, sú málokedy dostatočne zabezpečené.

Je veľa známych útokov na tieto zariadenia. K najzaujímavejším patrí napríklad otváranie inteligentných zámkov. Tieto často slabo zabezpečené zámky posielajú heslo pre ich otvorenie v nezašifrovanej podobe, preto stačí heslo zachytiť a použiť na ich otvorenie. Ďalšou metódou na otvorenie inteligentných zámkov je navodenie chybového stavu (zámku sa zašle veľa nevalidných správ). Zámok sa potom v chybovom stave sám odomkne. Ďalšou zraniteľnosťou využívanou útočníkmi je chýbajúce šifrovanie. Túto zraniteľnosť využil napríklad Mike Ryan, ktorý odpočúval monitor srdca. Na konci roku 2017 bol odhalený súhrn chýb a zraniteľností v Bluetooth známy pod menom Blueborne. Obsahuje popis chýb implementácií v Bluetooth knižniciach, ktoré boli používané na ovládanie Bluetooth čipu v zariadeniach. Síce nesúvisí priamo s komunikačným protokolom BLE, ale potvrdzuje veľkú zraniteľnosť týchto zariadení. Ako je zrejmé, zraniteľnosť BLE zariadení je veľkým problémom a bolo by potrebné viac dbať na bezpečnosť týchto zariadení.

Neexistuje nástroj, ktorý by dokázal bezpečnosť bluetooth zariadení klasifikovať a určiť mieru zraniteľnosti zariadenia. Existuje iba nástroj Ubertooth One, ktorý zostrojil Mike Ryan. Ubertooth One sa používa na odchyt a sledovanie paketov ako takzvaný snifer[17].

Tento nástroj je pasívny a dokáže len zachytiť Bluetooth a BLE pakety. Práve nástroj Ubertooth One bol použitý na odchyt paketov pri vypracovaní tejto bakalárskej práce.

Cieľom tejto baklárskej práce je navrhnúť nástroj na detekciu miery zabezpečenia BLE zariadení a vytvoriť systém na monitorovanie aktivity BLE zariadení nachádzajúcich sa v okolí. Používateľovi by mal systém poskytnúť informácie o miere zabezpečenia zariadenia. Pri monitoringu okolia by mal používateľa informovať o prípadných nechcených alebo podozrivých aktivitách zariadení, ako je napríklad pripájanie sa jeho zariadení na iné zariadenia. Tento nástroj rieši problém s nemožnosťou určenia miery zabezpečenia zariadenia a problém s nemožnosťou monitorovania aktivít zariadení. Mal by zobrazovať informáciu o zabezpečenosti zariadenia v jednoduchej forme, zrozumiteľnej aj používateľom menej znalým problematiky. Zároveň by mal obsahovať podrobnejšie výpisy pre užívateľov orientujúcich sa v danej problematike a mal by byť schopný poskytnúť podrobné informácie o zariadeniach a ich aktivitách.

Nasledujúca kapitola (2) je venovaná komunikačnému protokolu BLE, kde je vysvetlené jeho základné fungovanie do takej miery, aby bolo možné pochopiť fungovanie tohto protokolu z hľadiska bezpečnosti. Dôležitá časť v tejto kapitole z hľadiska bezpečnosti je hlavne časť 2.8, kde je popísané, ako sa vyberie metóda párovania. Samotný popis jednotlivých metód párovania Bluetooth Low Energy je v časti 2.9. Tieto metódy majú priamy súvis so zabezpečenosťou komunikácie. S bezpečnosťou súvisí ďalšia kapitola 3, kde sú popísané známe bezpečnostné incidenty v Bluetooth Low Energy. Je to stručný popis zraniteľností tohto komunikačného protokolu. V kapitole 4 je popis zariadenia Ubertooth, ktoré je používané na odchyt paketov Bluetooth Low Energy. Kapitola 5 popisuje návrh detekčného a monitorovacieho nástroja vyvinutého v rámci tejto bakalárskej práce. Táto časť popisuje, čo bolo cieľom, teda aké vlastnosti by mali mať jednotlivé nástroje a prečo. Kapitola 6 sa naopak zaoberá implementáciou, čiže ako bol návrh realizovaný. Na záver, po návrhu a implementácii, nasleduje experimentovanie a testovanie, ktoré je spolu so vzorovými výstupmi popísané v kapitole 7. V neposlednom rade, kapitola 8 je zhodnotenie práce a úspešnosti riešenia.

Kapitola 2

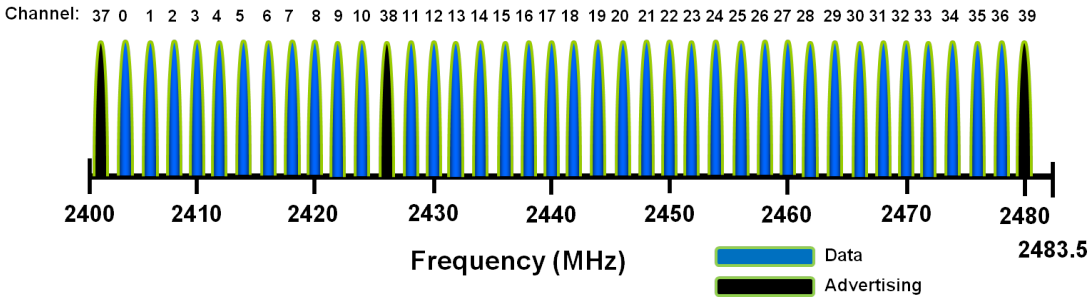
Bluetooth Low Energy

Táto časť popisuje fungovanie protokolu Bluetooth Low Energy. Popisuje hlavné princípy a mechanizmus pripojenia / prenášania dát. Nie je teda encyklopedickým prehľadom. Táto časť je zameraná na pochopenie fungovania BLE, a je teda výňatkom potrebných informácií z hľadiska bezpečnosti. Je rozdelená na viac častí. V prvej časti sú zhrnuté základné informácie o BLE, nasleduje časť popisujúca proces hľadania, pripájania, prípadného párovania a prenášania dát.

2.1 Základné informácie

Bluetooth Low Energy je bezdrôtová PAN (personal area network) technológia, ktorá je súčasťou Bluetooth od verzie 4.0. Oproti klasickému Bluetooth, je zameraná na nižšiu spotrebu energie, pričom má podobný bezdrôtový dosah. Bluetooth Low Energy protokol je vyvíjaný a štandardizovaný agentúrou Bluetooth Special Interest Group (SIG). Táto organizácia nepredáva zariadenia, len definuje a spravuje protokol[4].

Bluetooth je veľmi rozšírená technológia. Kúpiť si nový smartphone, tablet či notebook bez tejto technológie je prakticky nemožné. 100% mobilov, 100% tabletov a 100% notebookov predaných v roku 2018 malo túto technológiu [6]. Celkovo len za rok 2018 bolo predaných 2,1 miliardy takýchto zariadení s Bluetooth[6]. Aj preto je Bluetooth častou voľbou pre smart domácnosť. Na druhej strane sa predalo zhruba 140 miliónov smart wereables[5] *(fitness náramkov, hodiniek...)* a 650 miliónov smart home zariadení[7] *(zámkov, žiaroviek, termostatov...)*. Aj preto ju veľa výrobcov volí na komunikáciu s ich zariadením.



Obr. 2.1: Bluetooth Low Energy kanály [13]

Bluetooth Low Energy je rozšírením Bluetooth 4.x, pričom nie je kompaktibilná s Bluetooth Classic. Tak ako Bluetooth Classic využíva 2,4 GHz pásmo rozdelené na 40 kanálov.

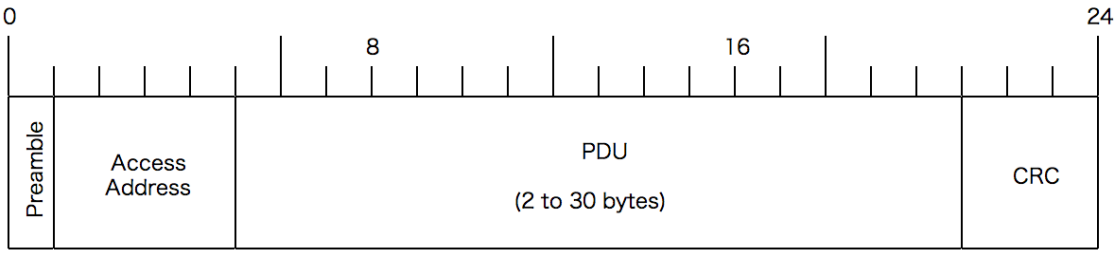
Medzi kanálmi sú vždy 2 MHz miesta a posledný kanál sa nachádza na 2485 MHz. Toto rozdelenie je vidieť na obrázku2.1. Oproti klasickému Bluetooth sú tieto kanály inak rozdelené. 3 kanály sú takzvané „Advertising Channels“. Tieto sú určené na vyhľadávanie zariadení, na pripojenie a prípadné spojenia typu broadcast. Laicky povedané na týchto kanáloch sa nepripojené slave zariadenie hlási. BLE podporuje spojenie 1:N, teda jeden Master (napríklad smartphone) môže mať napojených viac Slavov (napríklad žiaroviek, náramkov). Preto sa hlásia len nepripojené zariadenia. Na jednom z týchto kanálov si dohodnú parametre spojenia a presunú sa na jeden z 37 takzvaných „Data Channels“. Sú určené na obojsmerný prenos dát. Zariadenia však nepoužívajú stále jeden a ten istý kanál ale existuje takzvané „frequency hopping“. Čo znamená, že skáču postupne po rôznych kanáloch[13].

Frequency hopping je vykonávaný len na 37 dátových kanáloch, pričom sa riadi vzorcom 2.1:

$$f_{n+1} = (f_n + hop) mod 37 \tag{2.1}$$

Pričom f_{n+1} je nový kanál a f_n je pôvodný kanál. Hop je hodnota v rozmedzí od 5 do 16, ktorú si definujú na začiatku spojenia. Na konci je táto hodnota predelená celočíselne 37, čo je počet dátových kanálov. Môžeme vidieť, že frequency hopping nie je vôbec zložité. Rozdelenie kanálov na Advertising kanály a Data kanály, jednoduchý frequency hopping, pomáhajú väčšej výdrži batérie. Je dobré spomenúť, že zariadenie skáče len po vopred dohodnutých kanáloch. Túto informáciu si prenesú pomocou „Channel Map“ spolu s frequency hopping pri zahajovaní spojenia.

Základná štruktúra paketu je na obrázku 2.2 V tomto pakete sú hodnotami *Preamble*,



Obr. 2.2: Bluetooth paket [15]

hodnota určená na interné účely. *Access Address*, toto je adresa spojenia. Zariadenia, ktoré nie sú spojené, používajú adresu *0x8e89bed6*. Po úspešnom spojení sa vygeneruje nová voľná adresa. Táto adresa je potrebná pre komunikáciu s daným spojeným zariadením, keďže na jednom kanáli môže byť naraz viacero zariadení. Nasleduje PDU, ktorý sa mení v závislosti na type paketu. Posledný je kontrolný súčet.

2.2 Proces komunikácie

Nasledujúce kapitoly popisujú priebeh vytvárania spojenia a spojenie samotné. Na začiatku sa zariadenia musia vyhľadať, následne sa k sebe pripoja. Potom sa môžu spárovať. Pri párovaní sa vyberie metóda párovania, čo má spolu s verziou BT veľký vplyv na bezpečnosť.

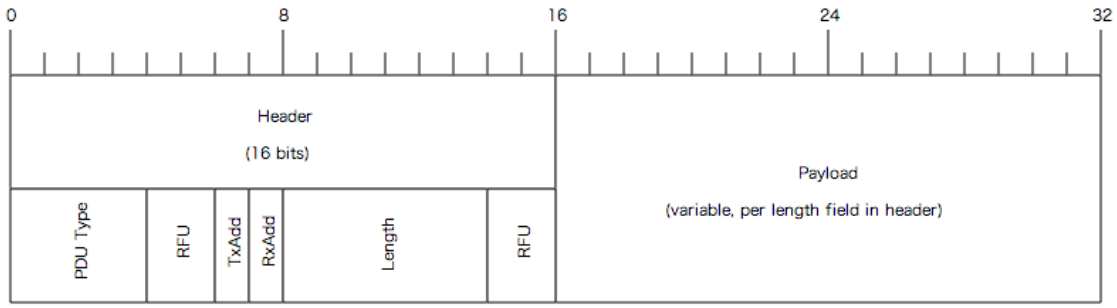
2.3 Advertising Process

Preto aby zariadenia sa mohli spojiť, musia sa najprv nájsť. Slave, ktorý nie je pripojený na žiadne zariadenie, vysiela v pravidelnom intervale Advertising pakety. Sú to vlastne pakety, ktorými dáva najavo, že je pripravený na spojenie [12]. Existujú 4 druhy Advertising paketov[15]:

- **General** - Undirected Connectable Advertising, posiela sa všetkým zariadeniam na advertising kanále a žiada o spojenie.
- **Directed** - Directed Connectable Advertising, posiela sa konkrétnemu zariadeniu, pričom zvyčajne hneď po tejto správe nasleduje pripojenie. Posiela sa ak sa zariadenia už poznajú, alebo napríklad pri výpadku siete. Tak isto žiada o spojenie.
- **Nonconnectable** - Undirected Non-Connectable Advertising, posiela sa všetkým zariadeniam, a nežiada o spojenie, ale o broadcast. Toto sa stáva jediným možným spojením pre zariadenia, ktoré sú len vysielačmi.
- **Discoverable** - Undirected Scannable Advertising, posiela sa všetkým a ide o zariadenia, ktoré je možné len oscanovať, ale nie pripojiť

Najčastejšie sa stretneme so všeobecným typom spojenia, t. j. *General*. Časový interval medzi jednotlivými odoslaniami advertising paketov je koštanta v rozmedzí od 20 milisekúnd až po 10,24 sekundy. Presne je špecifikovaná výrobcom[12].

Advertising paket je znázornený na obrázku 2.3. Tento paket obsahuje v hlavičke položku



Obr. 2.3: Advertising paket[15]

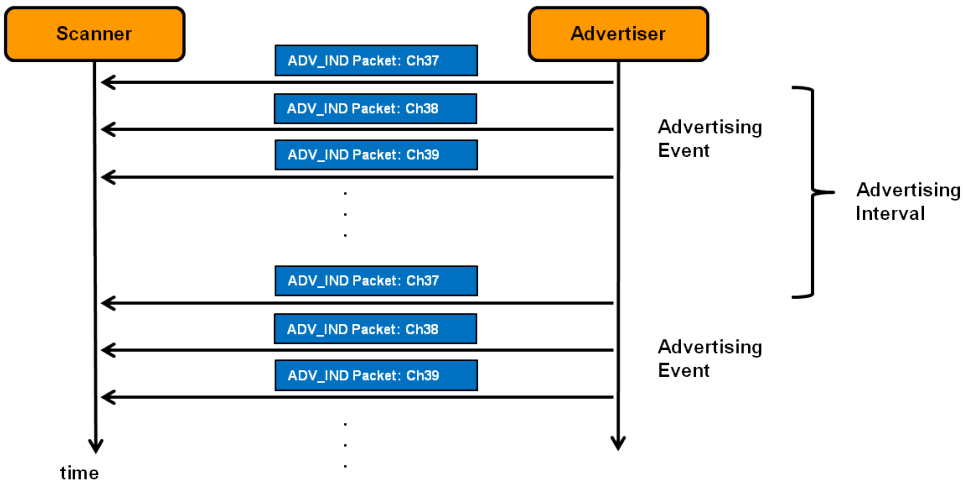
PDU_Type, ktorá obsahuje typ advertising paketu popísaného v sekcii 2.3. Tento môže nadobúdať hodnoty:

- **ADV_IND** (0000) - General
- **ADV_DIRECT_IND** (0001) - Directed
- **ADV_NONCONN_IND** (0010) - Nonconnectable
- **ADV_SCAN_IDN** (0110) - Discoverable

2.4 Scanning Process

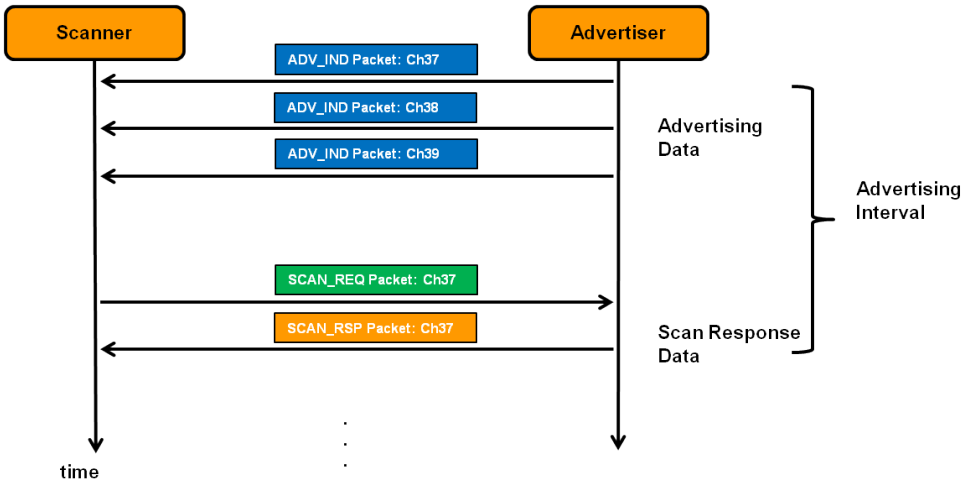
Pomocou procesu *Scanning* dokáže Master odhaliť zariadenia pre následné spojenie. Existujú dva typy scanovania. Aktívne a pasívne. Pri pasívnom scanovaní Master počúva na

Advertising kanáli a čaká na advertising pakety od Slavov. Toto scanovanie je znázornené na obrázku 2.4.



Obr. 2.4: Pasívne scanovanie[12]

Aktívne scanovanie sa používa väčšinou vtedy, keď Master device chce viac informácii, než obsahuje paket *ADV_IND*. Pri aktívnom scanovaní posiela Master *SCAN_REQ* a Slave mu odpovie pomocou *SCAN_RSP*, ktorý obsahuje viac informácií a to napríklad meno zariadenia. Vďaka aktívnemu scanningu vidíme názvy dostupných bluetooth zariadení. Tento proces je znázornený na obrázku 2.5.



Obr. 2.5: Aktívne scanovanie[12]

Toto aktívne scanovanie je realizované pomocou paketu s rovnakou štruktúrou, ako bol Advertising paket popísaný v sekcii 2.3 na obrázku 2.3. Rozdiel je v *PDU_Type*, ktorý môže obsahovať hodnoty:

- **SCAN_REQ** (0011) - General
- **SCAN_RSP_IND** (0100) - Directed

Po tom, čo sa zariadenia nájdu, je možné spojenie.

2.5 Connection Process

Connection, alebo spojenie či pripojenie zariadenia, je realizované po tom, ako získal Master dostatok informácií od Slave, vrátane MAC adresy. Keď je rozhodnutý spojiť sa so zariadením, pošle mu paket typu *CONNECT_REQ*, ktorý je znázornený na obrázku 2.6. Ten

Payload		
InitA (6 octets)	AdvA (6 octets)	LLData (22 octets)

Obr. 2.6: *CONNECT_REQ* paket[3]

obsahuje `InitA`, čo je adresa zariadenia, ktoré inicializuje spojenie (Master). Ďalej obsahuje `AdvA`, čo je adresa Slave zariadenia, na ktoré sa chce Master pripojiť. V `LLData` sú obsiahnuté informácie potrebné pre nadviazanie spojenia. Formát dát prenesených vrámci `LLData` je znázornený na obrázku 2.7.

LLData									
AA (4 octets)	CRCInit (3 octets)	WinSize (1 octet)	WinOffset (2 octets)	Interval (2 octets)	Latency (2 octets)	Timeout (2 octets)	ChM (5 octets)	Hop (5 bits)	SCA (3 bits)

Obr. 2.7: Formát dát `LLData` v rámci paketu *CONNECT_REQ* [3]

Tieto dáta prenesené vrámci `LLData` obsahujú hlavne nasledujúce informácie[3]:

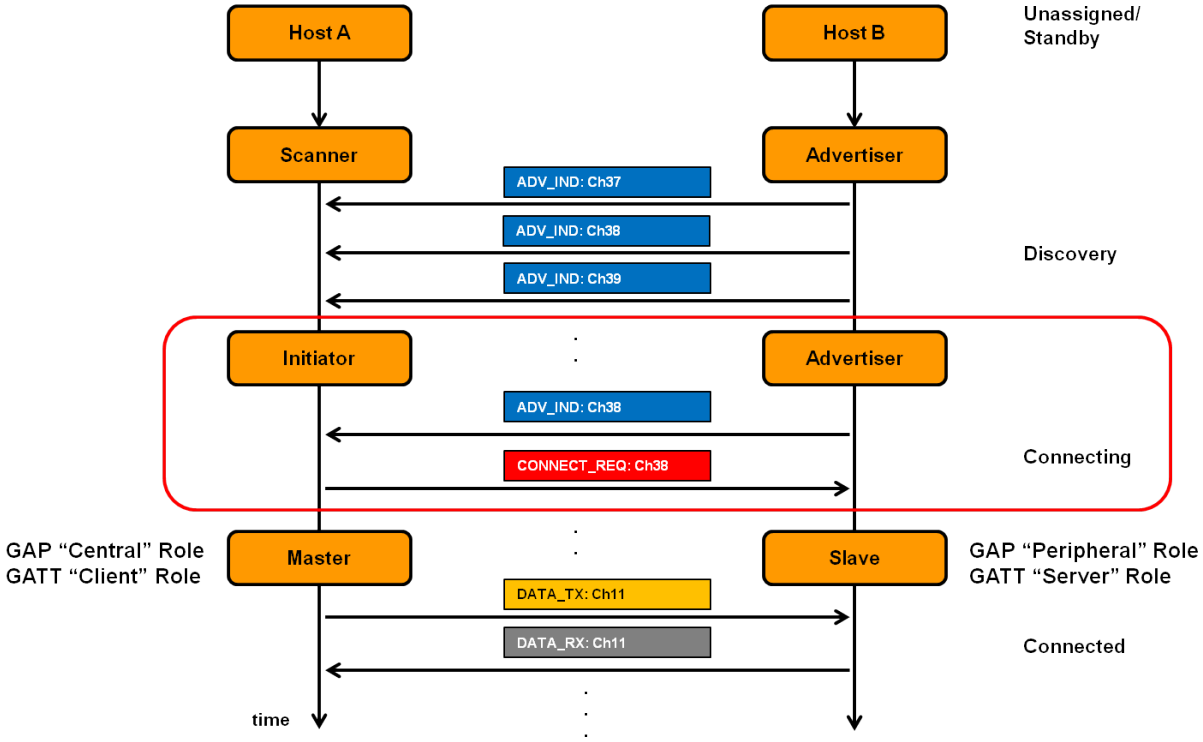
- **Frequency Hopping** - konštanta pre skákanie po kanáloch. Hodnota *hop* vo vzorci

$$f_{n+1} = (f_n + hop) mod 37 \tag{2.2}$$

- **Connection Interval** - čas medzi jednotlivými *connection event-mi*, popísané nižšie 2.6.
- **Slave Latency** - počet po sebe idúcich správ z eventov, ktoré Slave nemusí naslúchať (sú preskočené len ak nie je čo poslať)
- **Supervision Timeout** - maximálny čas medzi dvomi doručeniami validných dát. Po prekročení tohto času je spojenie považované za stratené.
- **Channel Map** - obsahuje kanály, pomocou ktorých zariadenia budú komunikovať. Teda na ktorých budú skákať.

Hneď ako je *CONNECT_REQ* paket odoslaný / doručený, je možné posielať dátové pakety. Priebeh pripojenia je znázornený na obrázku 2.8. Na tomto obrázku sú na začiatku zariadenia označené ako `Host A` a `Host B`. Ak zariadenie počúva na `advertising` kanáli, označujeme ho ako `Scanner`. `Scanner` počúva `advertising` pakety odoslané zariadením ktoré nazývame `Advertiser`. Ten vysiela `advertising` pakety na `advertising kanáloch`,

teda konkrétne na kanáli 37, 38 a 39. Ak `Scanner` odošle `CONNECT_REQ` paket, stáva sa initiátorom spojenia. Po nadviazaní spojenia sa tieto zariadenia presunú na dátové kanály, a majú ujasnené role v tomto spojení (teda master a slave). Z obrázku 2.8 si možno všimnúť, že na `CONNECT_REQ` Advertiser nijak neodpovedá, nie je potrebný súhlas či potvrdenie od zariadenia. Pri zlyhaní Master zistí, že na datovom kanáli nemá Slave, a pokus o pripojenie môže zopakovať.



Obr. 2.8: Schéma priebehu vzniku spojenia[11]

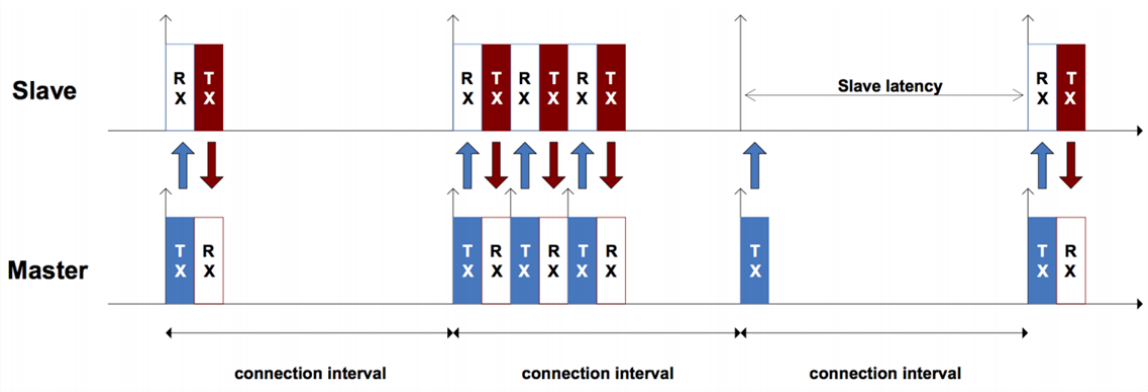
2.6 Dátové pakety

Po tom ako sa dve zariadenia spoja pomocou `CONNECT_REQ` paketu, komunikujú pomocou dátových paketov. Komunikujú pravidelne, a to aj vtedy, ak nemajú dáta. Interval tejto komunikácie je daný pri vytváraní spojenia parametrom *Connection interval*. Connection interval je v rozmedzí od 7,5 ms do 4 s. Ak nie sú dáta na prenesenie, PDU má veľkosť 0 bitov. Vtedy ide len o udržanie spojenia a zariadenie takto môže včas detekovať stratu spojenia. Tento connection event je podrobne znázornený na obrázku 2.9

Bluetooth Low Energy poskytuje spoľahlivé spojenie. Obsahuje CRC hodnotu, ktorá kontroluje poškodenie paketu a jeho celistvosť. Ak je nejaký paket poškodený, alebo sa „stratí“, požiada sa o znovu-odoslanie. Tento paket je znovu odoslaný v ďalšom connection evente. O toto znovu-odoslanie sa žiada až dovtedy, kým nie je paket korektne doručený.

Podľa prvého bitu sa dátové pakety delia na 3 kategórie:

- 0x1 - prázdny paket pre udržanie spojenia alebo pokračovanie správy v predchádzajúcom pakete



Obr. 2.9: Connection event[11]

- 0x2 - nový paket, respektíve správa, pre nás podstatná hodnota ďalších bajtov je 0x0006, ktorá indikuje párovací paket
- 0x3 - ovládací paket, ktorý slúži napríklad na zmenu kanálov, po ktorých skáče alebo na požiadavok na začatie šifrovania
  - nadobúda hodnoty od 0x00 do 0x15. Medzi zaujímavé hodnoty patria napríklad:
  - 0x01 - LL_CHANNEL_MAP_REQ - aktualizácia channel map
  - 0x02 - LL_TERMINATE_IND - strata spojenia
  - 0x03 - LL_ENC_REQ - požiadavka na šifrovanie, kľúče
  - 0x05 - LL_START_ENC_REQ - požiadavka na začatie šifrovania
  - 0x0C - LL_VERSION_IND - voľba verzie Bluetooth (väčšinou zvolená na začiatku spojenia)

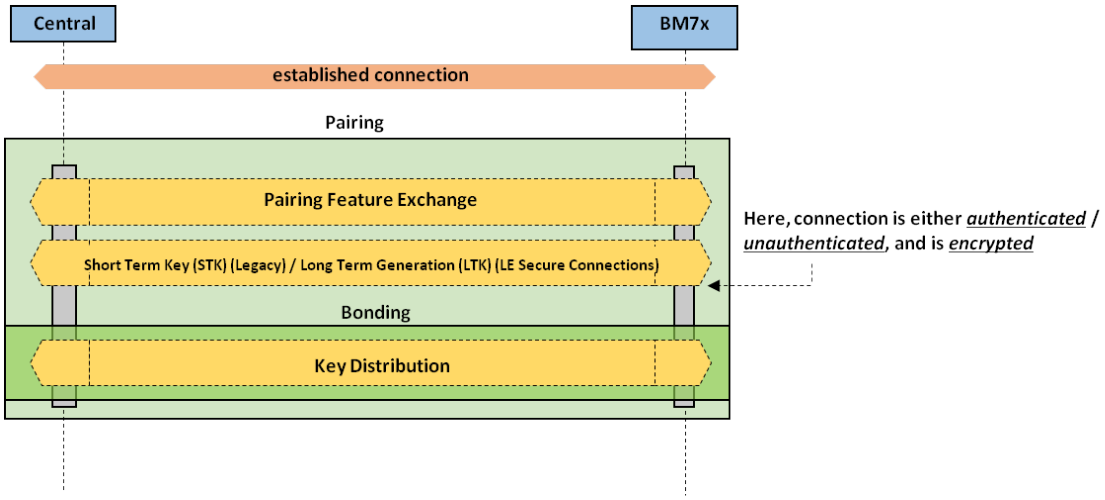
Po pripojení zariadenie môže požiadať o párovanie.

2.7 Párovanie a bonding

Párovanie a Bonding je vykonávané po tom, ako bolo uskutočnené spojenie. Na začiatku je vhodné vysvetliť tieto dva pojmy[1]:

Pairing pokiaľ dve zariadenia chcú spolu komunikovať bezpečnou cestou, musia sa spárovať. Väčšina BLE zariadení si bez párovania nebude vymieňať žiadne informácie, keďže by boli posielané ako „plain text“, teda holý text bez akéhokoľvek šifrovania. Tým pádom, útočníkovy by stačilo odchytiť len tento paket, na zistenie by všetkých informácií. Samozrejme existujú zariadenia, čo takto komunikujú. Môžu používať šifrovanie na aplikačnej úrovni, ale detekcia a analýza šifrovania na aplikačnej úrovni nebola predmetom riešenej bakalárskej práce. Ak chceme spojenie zabezpečiť, je potrebné párovanie. Párovanie je autentifikácia dvoch zariadení. Výmena *short-term key*(STK) a *long-term key*(LTK). Pomocou týchto kľúčov je šifrovaná ďalšia komunikácia. Typy šifrovania a bezpečnosti spojení sú spísané ďalej v kapitole 2.9.

Bonding je to uloženie kľúčov pre ich neskoršie použitie. Uloženie kľúčov umožňuje rýchle znovupripojenie zariadenia bez potreby výmeny kľúčov. Či si tieto kľúče vymenia alebo nie, je dané *bonding* flagom, ktorý je súčasťou párovacieho paketu.



Obr. 2.10: Proces párovania a Bonding[14]

Proces párovania a Bonding je znázornený na obrázku 2.10. Na obrázku 2.10 možno vidieť, že párovanie sa uskutoční až po spojení zariadení. „Pairing Feature Exchange“ je párovací paket. Na základe tohto paketu sa vygeneruje STK (short term key alebo dočasný kľúč), ktorý sa použije na zašifrovanie LTK (long term key alebo stály kľúč), ktorý si taktiež vymenia. Ak si kľúč uložia pre budúce použitie, nazýva sa to Bonding.

Párovací paket obsahuje informácie, ktoré budú neskôr použité pri výbere metódy spojenia. Jeho presná štruktúra je znázornená na obrázku 2.11. Nasleduje krátky popis

Field	Code	IO	OOB	AuthReq (1 Byte)					Maximum	Initiator	Responder
Sub-define	(1 Byte)	Cap (1 Byte)	DF (1 Byte)	BF	MITM	SC	KP	Reserved	Encryption Key Size (1 Byte)	Key Distribution (1 Byte)	Key Distribution (1 Byte)
Bits*	8	8	8	2	1	1	1	3	8	8	8

Obr. 2.11: Párovací paket[1]

hodnôt, prenášaných v tomto pakete 2.11. Párovací paket obsahuje na prvom bajte hodnotu „Code“, pričom táto hodnota indikuje:

- 0x01 - Pairing Request - odosiela ho Master, ktorý je iniciátorom párovania
- 0x02 - Pairing Response - odpoveď Slave

Ďalej obsahuje hodnotu „IO Cap“, čo indikuje jeho vstupno-výstupné rozhrania. Popis možných vstupno-výstupných zariadení je súčasťou prílohy A spolu s hodnotami, ktoré može „IO Cap“ nadobúdať. Táto hodnota je podstatná v poslednom kroku voľby metódy spojenia popísaných nižšie.

Ďalšou hodnotou je „OOB“, čo značí výmenu informácií inou cestou, ako je Bluetooth LE. Toto môže byť napríklad QR kód, alebo technológia NFC.

Následuje „BF“ teda Bonding flag, ktorá určuje, či si zariadenia uložia svoje kľúče pre ďalšie spojenie. „MITM“ flag definuje, či zariadenie požaduje Man-In-The-Middle ochranu. „SC“ značí Bluetooth Secure Connection. Je to typ spojenia, ktorý je lepšie zabezpečený. Tieto hodnoty sú potrebné pre výber motódy spojenia.

2.8 Voľba metódy spojenia

Po výmene informácií v párovacích paketoch nastane voľba metódy spojenia, ktorá má veľký vplyv na zabezpečenie. Táto voľba sa vykoná v 4 krokoch [2].

Prvým krokom je skontrolovanie SC flagu z oboch párovacích paketov. Ak je na oboch zariadeniach SC nastavený na 1, pokračujeme krokom 2. Inak sa krok 2 preskakuje a pokračujeme krokom 3.

V druhom kroku je rozhodnuté, že spojenie bude typu LE Secure Connection. Riadime sa ďalej tabuľkou 2.1. Ak aspoň jedno zariadenie má možnosť použitia OOB je primárne použité. Ak nie, skontroluje sa hodnota MITM. Ak aspoň jedno zariadenie obsahuje ochranu proti MITM útokom, pokračujeme štvrtým krokom, inak sa použije metóda `JustWorks`. V treťom kroku je rozhodnuté, že spojenie bude typu LE Legacy pairing. Riadime sa ďalej

		Master			
		OOB flag 1	OOB flag 0	MITM flag 1	MITM flag 0
Slave	OOB flag 1	Použi OOB	Použi OOB		
	OOB flag 0	Použi OOB	Skontroluj MITM		
	MITM flag 1			Krok 4	Krok 4
	MITM flag 0			Krok 4	Použi JustWorks

Tabuľka 2.1: Tabuľka 2. kroku pri výbere metódy párovania

podobnou tabuľkou 2.2. Ak obe zariadenia poskytujú možnosť použitia OOB je primárne použité. Ak nie, skontroluje sa hodnota MITM. Ak aspoň jedno zariadenie obsahuje ochranu proti MITM útokom, pokračujeme štvrtým krokom, inak sa použije metóda `JustWorks`.

		Master			
		OOB flag 1	OOB flag 0	MITM flag 1	MITM flag 0
Slave	OOB flag 1	Použi OOB	Skontroluj MITM		
	OOB flag 0	Skontroluj MITM	Skontroluj MITM		
	MITM flag 1			Krok 4	Krok 4
	MITM flag 0			Krok 4	Použi JustWorks

Tabuľka 2.2: Tabuľka 3. kroku pri výbere metódy párovania

Vo štvrtom kroku sa rozhodne o metóde spojenia pomocou vstupno-výstupných zariadení. Pre zložitosť je tabuľka súčasťou prílohy B. V tomto konečnom kroku sa rozhodne o jednej z metód spojenia popísaných v časti 2.9.

2.9 Popis párovacích metód

Bluetooth Low Energy poskytuje dve základné metódy spojenia. Prvá metóda spojenia LE Legacy Pairing je stará metóda spojenia Bluetooth. Je veľmi zraniteľná a kľúče sú veľmi ľahko odhaliteľné. A to vďaka tomu, že TK (Temporary Key) je odoslaný ako plain text, ním je následne zašifrovaný STK (Short Term Key), ktorý sa používa na šifrovanie komunikácie. Ak útočník počúva od začiatku komunikácie, má všetky potrebné kľúče k jej odšifrovaniu. Všetky typy spojenia až na OOB (kde sa TK prenesie iným spôsobom) sú teda nebezpečné. Obsahuje 3 spôsoby:

- **Just Works** - z hľadiska bezpečnosti najhorší a najmenej zabezpečený spôsob spojenia aký má BLE. TK sú samé nuly. Nezabezpečené proti aktívnym a ani pasívnym útokom. Je však najľahší na zhotovenie.
- **Passkey** - pre chybnú výmenu kľúčov nechráni proti pasívnym útokom. Chráni však proti MITM útokom.
- **Out Of Band** - bezpečné, kľúče sa vymenia inou cestou/technológiou. Bezpečnosť použitej technológie je však otázna.

Druhá metóda spojenia je LE Secure Connections. Po kritike odbornej verejnosti, vzhľadom na chybnú nezabezpečenú výmenu kľúčov, Bluetooth vytvoril nové možnosti spojenia. Obe zariadenia pre tento nový typ spojenia musia podporovať minimálne verziu Bluetooth 4.2. Na výmenu kľúčov tentokrát zvolili Elliptic Curve Diffie-Hellmann, ktorý je bezpečný. Obsahuje 4 spôsoby:

- **Just Works** - ochráni proti pasívnym útokom, keďže kľúče už nie je možné odhaliť. Neochráni však proti aktívnym útokom. Tieto zariadenia nemajú väčšinou vstupno-výstupné rozhrania.
- **Numeric Comparison** - chráni proti pasívnym a aj proti aktívnym útokom.
- **Passkey** - chráni proti pasívnym a proti MITM útokom.
- **Out Of Band** - bezpečné, kľúče sa vymenia inou cestou / technológiou. Bezpečnosť použitej technológie je však otázna.

Kapitola 3

Známe bezpečnostné incidenty v BLE

Táto časť popisuje bezpečnostné incidenty v BLE a to do miery potrebnej pre pochopenie útokov na BLE. Pre ešte podrobnejšie informácie odporúčam pozrieť sa do odbornej literatúry. Jednotlivé výskumné skupiny a bezpečnostné firmy vykonali veľa podobných útokov. Preto si jednotlivé útoky, ktoré a odlišujú zväčša použitými technológiami alebo napadnutými zariadeniami, zhrnieme viac vo všeobecnosti. V závere tejto kapitoly 3.5 je spísané zhrnutie, ktoré je dostatočné pre rýchle zorientovanie v zraniteľnostiach BLE.

Útoky na BLE sú celkom bežné. Vďaka veľkým chybám v protokole do verzie 4.2 sú veľmi ľahko vykonateľné. Jeden z prvých útočníkov je Mike Ryan, ktorý v roku 2011 pre svoje výskumné účely vyvinul Ubertooth One, popísaný v časti 4. So svojím prvým útokom s názvom „Hacking Bluetooth Low Energy: I Am Jack's Heart", ktorý predstavil na konferencii ToorCon v roku 2012, spôsobil vlnu útokov na Bluetooth Low Energy. Značne sa podpísal pod pochybnosti o bezpečnosti BLE. Ostatné výskumné skupiny často ťažia z jeho znalostí.

Informácie popísané v tejto časti čerpajú z literatúry [9][8][10][16].

3.1 Replay útok

Je to jeden z najbežnejších útokov. Priebeh útoku je nasledovný:

1. Útočník si odchytáva komunikáciu medzi dvoma zariadeniami. Tú si zároveň ukladá. Toto je možné vďaka Ubertoothu One 4, ktorý umožňuje snifovanie komunikácie a uloženie si tejto komunikácie vo viacerých formátoch.

2. Keď sa Slave vzdiali od Mastera, útočník vyhľadá zariadenie a pripojí sa na zariadenie.

3. Následne útočník prehrá dáta z uloženého súboru. Buď všetky alebo vybrané príkazy. Ak je zariadenie slabo zabezpečené, bude reagovať a príkazy vykoná.

Tento typ útoku veľmi dobre funguje pri zariadeniach, ktoré nepoužívajú šifrovanie. Ale ani použité šifrovanie nie je zárukou bezpečnosti. Komunikácia môže byť riadne zašifrovaná, ale pokiaľ neobsahuje mechanizmus proti Replay útokom, aplikácia zakaždým vygeneruje rovnakú zašifrovanú komunikáciu. Aj keď je tento mechanizmus útoku pomerne jednoduchý, dajú sa s ním prekonať ochrana dokonca niektorých Smart zámkov.

3.2 Fuzzing

Fuzzing je útok, pri ktorom sa snažíme vyvolať neočakávaný stav zariadenia. Pokiaľ to zariadenie nemá dostatočne ošetrené, môže zmeniť svoj stav. Niektoré smart zámky sa pri neočakávanom stave otvorili.

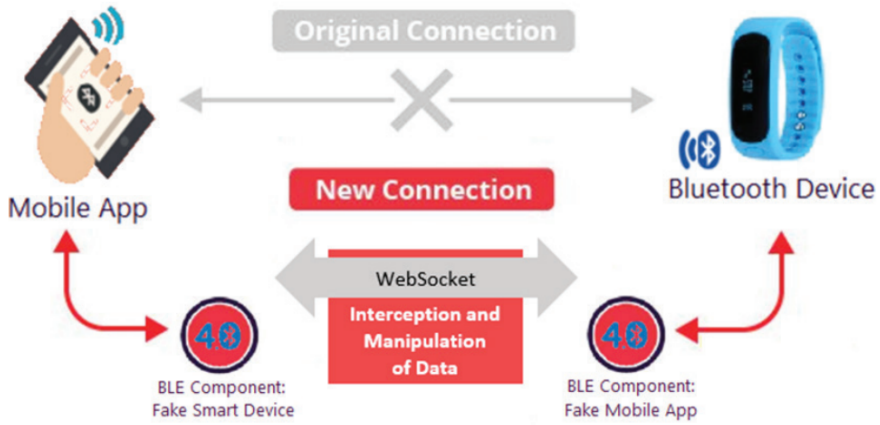
3.3 MAN-IN-THE-MIDDLE

Man-In-The-Middle je útok, pri ktorom útočník „sedí“ medzi dvoma komunikujúcimi zariadeniami. Klasické MITM funguje následovne:

1. Keď jedna strana (napríklad klient) odošle dáta, útočník pôsobí ako príjemca.
2. Následne si útočník môže tieto dáta upraviť a odoslať upravené druhej strane.
3. A naopak, keď druhá strana (napríklad server) pošle dáta, správa sa ako klient a môže napríklad upravené dáta poslať druhej strane.

To sa však nedá úplne uplatniť pri BLE, a to kvôli typickému obmedzeniu technológie Bluetooth, pre ktoré zariadenie môže byť pripojené iba k jednej strane a preto sa nemôže pripojiť na oba konce súčasne.

Pre MITM útok pri BLE sú potrebné dve BLE zariadenia, ktoré fungujú súčasne a každé z nich komunikuje s jednou stranou. Tieto dve BLE zariadenia musia medzi sebou komunikovať, na čo môže byť použitá napríklad aplikácia WebSocket. Táto umožňuje obojsmernú komunikáciu medzi týmito dvomi zariadeniami. Táto schéma útoku je znázornená na obrázku 3.1.



Obr. 3.1: Man-In-The-Middle

Útočník má odchytený LTK vďaka chybnému preneseniu TK, ktorý je prenesený v otvorenej podobe a ten použije na komunikáciu, alebo TK skúsi odhadnúť. Pri najčastejšej metóde spojenia Just Works je to 0. Ak použije PassKey Entry, môže sa pokúsiť tento kľúč odhadnúť hrubou silou (teda tipovaním keďže kombinácií je len milión). Tieto útoky sú časté do verzie 4.1 keďže pri verzii 4.2 začal bluetooth používať správnu výmenu kľúčov a je možné ho považovať za bezpečný.

3.4 Čítanie dát

Niektoré zariadenia nepoužívajú šifrovanie. V tomto prípade je postačujúce odchytiť komunikáciu a prečítať dáta. Takto vieme zistiť napríklad heslo k zámku či stav žiarovky.

Aj v prípade šifrovania sa dáta dajú zistiť, a to v prípade, ak je použité JustWorks alebo PasskeyEntry.

Pred vytvorením šifrovanej relácie si master a slave musí vytvoriť (LTK). Typicky master a slave vytvoria LTK, uložia ho a znova ho použijú v budúcnosti. V opačnom prípade, si tento kľúč master a slave vytvorí pomocou „key exchange protocol“ (protokolu o výmene kľúčov). „Key exchange protocol“ začína výberom TK, 128-bitový kľúč AES, ktorého hodnota závisí na metóde spojenia. Po výbere TK si ho potvrdia takzvanou potvrdzovacou hodnotou. Všetky hodnoty na výpočet tejto hodnoty sa prenesú ako plain text. Teda okrem LK, ktorý sa z týchto hodnôt dá vypočítať. TK má pri JustWorks hodnotu 0 a pri PassKey Entry má hodnotu 0 - 999 999. Pre PassKey entry brute force attack nám postačuje menej ako sekunda na prelomenie. Po potvrdení vypočíta master a slave STK, ktorý prenesie a zašifruje pomocou TK. Následne je LTK zašifrovaný STK. Vďaka tejto chybnej výmene kľúčov je útok celkom ľahko vykonateľný.

3.5 Zhrnutie útokov

Po naštudovaní známych bezpečnostných incidentov je jasné, že všetky bezpečnostné problémy spája absencia šifrovania, či dokonca párovania zariadení. BLE protokol síce disponuje AES-128 šifrovaním, no toto šifrovanie je voliteľné, preto veľa výrobcov toto šifrovanie nepoužije a posiela informácie ako text v otvorenej podobe. Toto bohužiaľ platí pre väčšinu smart zariadení v súčasnosti. Do verzie 4.2 bola chybná výmena kľúčov pri párovaní. Od verzie 4.2 bol pridaný Elliptic-curve Diffie–Hellman protokol pre výmenu kľúčov, ktorý je bezpečný, ale jeho použitie je opäť voliteľné (na základe šifrovania). Z dôvodu čo najmenších výrobných nákladov, nie sú tieto bezpečnostné prvky implementované. Čo využívali všetky útoky na BLE zariadenia. Uveďme si teda dva základné princípy, ktoré umožnily známe útoky na BLE:

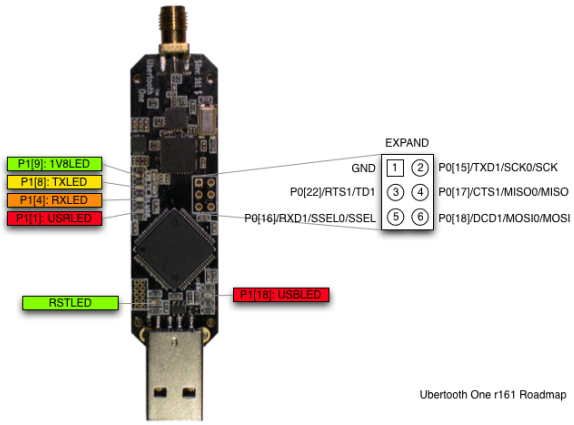
- **Plain text** - odosielanie dát ako holý text bez použitia šifrovania
- **Chybná výmena kľúčov** - aj napriek použitému šifrovaniu je možné komunikáciu odšifrovať. Viac v časti 3.4.
- **Nedostatočná ochrana proti replay útokom** - možnosť prehrania rovnakého príkazu z útočníkového zariadenia.
- **Out Of Band** - bezpečné, kľúče sa vymenia inou cestou/technológiou. Bezpečnosť použitej technológie je však otázna.

Kapitola 4

Ubertooth One

Ubertooth One je open-source 2.4 GHz bezdrôtová vývojárska platforma, určená pre experimentovanie s Bluetooth. Predáva sa za zlomkovú cenu oproti komerčným monitorovacím platformám. Ubertooth One bol vyvinutý v roku 2011 výlučne pre BT a BLE potreby. Jeho vývojár Mike Ossmann zahájil jeho vývoj keď si uvedomil, že neexistuje žiadny adaptér BT, ktorý by ponúkal potrebné schopnosti[17].

Prístroj je navrhnutý predovšetkým ako zdokonalený prijímač s rozhraním Bluetooth. Ponúka možnosti nad rámec tradičných adaptérov, ktoré umožňujú jeho používanie ako monitorovacej a sledovacej platformy BT. Hoci je hardvér zariadenia vhodný na vysielanie signálu, firmvér v súčasnosti podporuje iba príjem a minimálne funkcie odosielania advertising paketov[17].



Obr. 4.1: Ubertooth One[17]

Ubertooth One je postavený na mikrokontroléri ARM Cortex-M3 a je schopný zachytiť a demodulovať signály v pásme ISM 2,4 GHz s úzkou šírkou pásma len 1 MHz. Toto je vhodné na odchyt paketov klasického Bluetoothu a Bluetooth Low Energy[17].

Vlastnosti:

- 2.4 GHz vysielanie a prijímanie, na tejto frekvencii vysiela BLE
- štandartný Cortex Debug Connector (10-pin 50-mil JTAG), umožňuje ladenie a sledovanie adaptéra
- In-System Programming (ISP) sériový konektor, pre schopnosť nainštalovať alebo aktualizovť firmvér

- 6 indikačných lediek indikujúcich stav zariadenia

Komponenty:

- RP-SMA RF connector - štandartný konektor pre anténu, anténu je možno vymeniť za inú, prípadne smerovú
- CC2591 RF front end - mikročip
- CC2400 wireless transceiver - čip na vysielanie na 2,4 GHz pásme
- LPC175x ARM Cortex-M3 microcontroller with Full-Speed USB 2.0 - mikročip na riadenie zariadenia a komukáciu s PC
- USB A plug - vstupno-výstupné zariadenie

Ubertooth obsahuje repozitár, ktorý obsahuje viacero programov. Umožňuje napríklad zobraziť spektrálnu analýzu pásma, či zachytávať bluetooth pakety do PCAP súboru. Tie je možné následne prezerať pomocou programu Wireshark. Väčšina programov je vo fáze vývinu a nie sú úplne dokončené. Programy a ich stručný popis:

- **Spectrum analysis** - spektrálna analýza pásma (graf) na základe zachytenej komunikácie
- **Ubertooth-rx** - nástroj na odchytávanie komunikácie klasického Bluetooth
- **Ubertooth scan** - nástroj na vyhľadávanie Bluetooth zariadení
- **Ubertooth-btle** - nástroj pre BLE

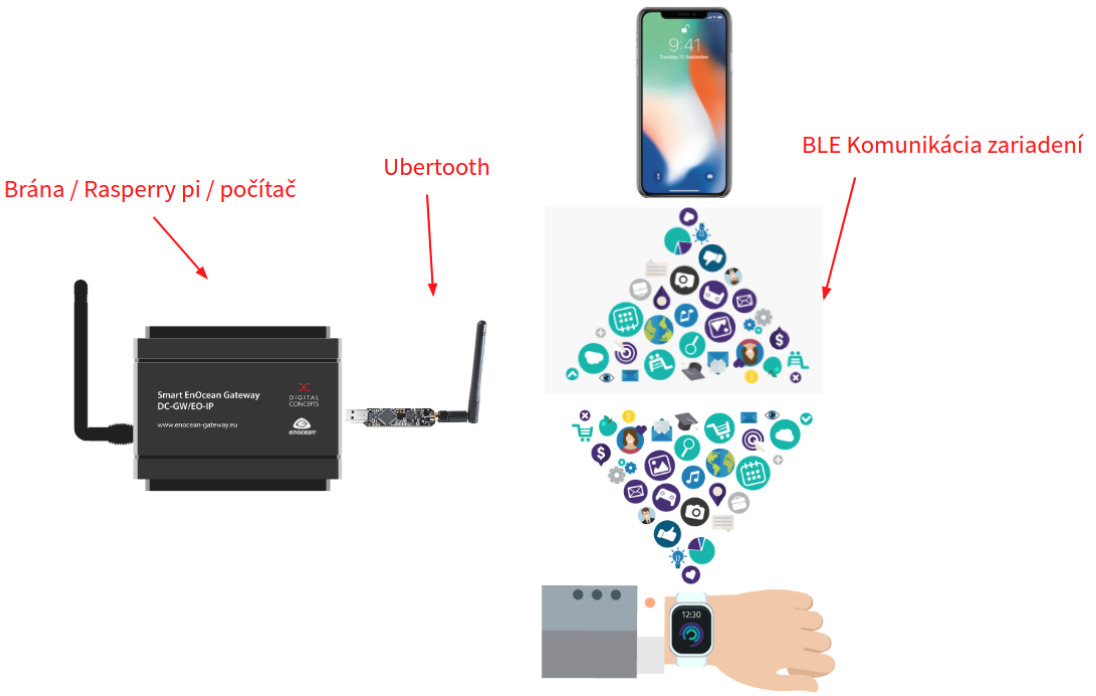
Pre potreby tejto bakalárskej práce je podstatný hlavne posledný program. Bohužial tento nástroj je označený ako „experimental“ a je skoro nefunkčný. Rušenie zariadenia, odosielanie dát na Advertising kanál a chytenie sa už prebiehajúceho spojenia, sa ukázali ako úplne nefunkčné. Na druhej strane odchytávanie BLE paketov po tom, ako bolo odchytené pripojenie zariadení na Advertising kanáli fungovalo, aj keď iba čiatočne. A to z dôvodu, že produkovalo navyše plno falošných paketov.

Nakoniec sa zariadenie ukázalo použiteľné len ako snifer, a to tým spôsobom, že nám zariadenie pošle pakety v hexadecimálnej podobe a tie následne spracujeme vlastným spôsobom.

Kapitola 5

Návrh detekčného a monitorovacieho nástroja

Po naštudovaní fungovania Bluetooth Low Energy a možností nástroja Ubertooth, bol vytvorený návrh riešenia. Tento návrh berie do úvahy obmedzenia nástroja Ubertooth a snaží sa ich minimalizovať. Boli definované požiadavky na výsledný program. V prvom rade bola navrhnutá architektúra. Podľa nej je nástroj Ubertooth umiestnený do brány prípadne do nejakého zariadenia typu Raspberry Pi či počítača. Na bráne či Raspberry Pi je spustený program vytvorený v rámci tejto bakalárskej práce, ktorý bude komunikovať s Ubertooth zariadením. Ak sa v blízkosti objavia Bluetooth Low Energy zariadenia Ubertooth túto komunikáciu odchytí. Ubertooth programu poskytuje odchytené dáta a program dáta spracuje a následne vykoná analýzu. Jednoduchá schéma tejto popísanej architektúry je znázornená na obrázku 5.1.



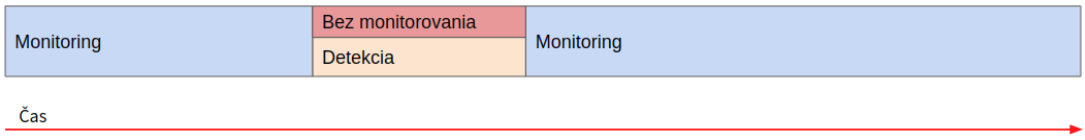
Obr. 5.1: Schéma návrhu architektúry

Výsledný návrh pracuje pasívnym spôsobom. Teda komunikáciu iba odchytáva a nič nevysiela. Pri BLuetooth Low Energy je najväčším problémom skákanie / menenie frekvencií. Ubertoooth je síce po hardvérovej stránke schopný odosielať dáta, avšak frekvencie sa menia príliš rýchlo, a preto býva problém komunikáciu odchytiť a nestratiť ju. Na vysielanie nezostáva časový priestor. Nevysielanie dát má svoje výhody, keďže útočník nebude mať možnosť zistiť prítomnosť tohto nástroja. A pre potreby bakalárskej práce je čisto vysielanie dát dostačujúce. Boli navrhnuté dve hlavné funkcionality výsledného programu. Program dokáže **detekovať** a **monitorovať** zariadenia v jeho okolí. Funkcionality detekovania a monitorovania sú popísané v ďalších častiach práce.

5.1 Detekcia a monitoring

Detekcia a monitoring sú dve základné funkčnosti výsledného nástroja. Detekcia má za úlohu detekovať, teda odhaliť bezpečnostné vady v Bluetooth Low Energy zariadeniach. Na druhej strane monitoring má za úlohu odpočúvať na `advertising` kanáli a tým monitorovať aktivitu zariadení v jeho okolí. Nástroj tak vie detekovať útočníka v okolí a jeho prípadné aktivity. Detekčný nástroj má za úlohu určiť úroveň zabezpečenia komunikácie zariadenia. Avšak v dnešnej dobe je väčšina zariadení slabo zabezpečená a je ťažké zaobstarať dobre zabezpečené zariadenia. A preto sa očakáva, že aj napriek tomu, že detekčný nástroj vyhodnotí zariadenie ako slabo zabezpečené, užívateľ si ho ponechá. V tom prípade použije monitorovací nástroj, ktorý ho bude informovať o pripájaní neautorizovaných zariadení na jeho zariadenia. Tieto dva navrhnuté nástroje sa svojou funkcionalitou dopĺňajú.

Najskôr sa uvažovalo o súbežnej funkcionalite týchto dvoch nástrojov. To znamená, že monitorovací nástroj by monitoroval `advertising` kanál. V prípade vzniknutia spojenia by sa spustil detekčný program. Ten by vykonal analýzu spojenia a po ukončení by sa vrátil do monitorovacieho módu. Zostaviť takýto program nebolo problémom. Nakoniec ale bolo od toho upustené, keďže Ubertooth dokáže odchytávať len jednu frekvenciu v jednom čase. Teda nemôže naraz odchytávať `advertising` kanál, kde sa vykonáva monitoring a `dátové` kanály, na ktorých sa odohráva detekcia. Útočník by môhol využiť chvíľu, kedy by monitorovanie nebolo vykonávané. Toto znázorňuje obrázok 5.2.



Obr. 5.2: Detekcia a monitoring pri 1. návrhu

Následne v návrhu boli tieto dve funkcionality rozdelené na samostatné programy. Súhlasí to aj s Unixovou logikou, kedy jeden program odzrkadľuje jednu funkcionalitu. V praxi si používateľ nebude až tak často kupovať nové zariadenia aby bolo potrebné spúšťať detekciu automaticky. Ocení skôr kvalitnejší monitoring, ktorý je lepšie vykonávať spoľahlivo ako nespoľahlivo spolu s detekciou. Bližšie informácie o monitoringu sú popísané v časti 5.2 a o detekcii v časti 5.3.

Pri návrhu bolo taktiež potrebné zvoliť vhodnú prácu so zariadeniami. A to v prvom rade s ich identifikovaním. Zariadenia možno identifikovať pomocou ich `MAC` adresy. Tá má tvar `c2:3e:a6:79:63:44` a pre bežného používateľa je, pochopiteľne, ťažko zapamätateľná. Preto bolo vhodné si zvoliť nejaký identifikátor. Identifikovanie zariadení postupne (t. j.

1,2,3...) sa samozrejme ponúkalo ako prvé, ale nebolo zvolené z nasledujúcich dôvodov. Je vhodné aby identifikátory mali rovnakú číselnú dĺžku. Hlavným dôvodom však bolo odobratie zariadenia. Napríklad, čo by sa stalo, ak by sme odobrali napríklad tretie zariadenie. Posunuli by sme číslovanie alebo by ostala medzera v číslovaní? Prípadne by nové zariadenie obsadilo túto medzeru? Preto bol zvolený štvormiestny číselný identifikátor. 4 cifry by mali dostatočne stačiť na označenie všetkých zariadení v okolí, pretože takto vieme označiť 10000 zariadení a pri väčšom počte zariadení by bolo komunikačné pásmo tak zahltené, že by nebolo možné komunikovať. Dlhší identifikátor by bol zároveň zle zapamätateľný pre používateľa. Ak je zariadenie zaznamenané po prvýkrát, vygeneruje sa mu jedinečný identifikátor, pomocou ktorého sa bude označovať v záznamoch. Tento identifikátor bude nemenný pre dané zariadenie po celú dobu kým používateľ toto zariadenie nezmaže zo zoznamu.

5.2 Detekčný nástroj

Detekčný nástroj alebo detektor má za úlohu odhaliť bezpečnostné problémy v zariadení. Klasický prípad použitia je v prípade, ak si používateľ kúpi nové zariadenie, zostavuje si `Smart Home`, či sa rozhoduje dvoma zariadeniami. Vtedy používateľ spustí detekčný nástroj, pripojí zariadenia (či na bránu alebo smartphone...) a následne si overí mieru ich bezpečnosti. DetekČný nástroj klasifikuje úroveň bezpečnosti zariadenia a s touto informáciou používateľ následne môže pracovať.

Z hľadiska funkčnosti je nástroj Ubertooth nastavený na `advertising` kanál. V prípade zachytenia `connect` paketu sa spolu so zariadeniami pohybuje po `dátových kanáloch`. Na týchto dátových kanáloch si zariadenia vymieňajú informácie o spojení, ako je verzia Bluetooth, či zapnutie šifrovania. Veľa podstatných dát je poslaných v `párovacích` paketoch. Tie si takisto posielajú dátovými kanálmi, sú tvorené dvojicou: požiadavka (request) a odpoveď (respond). Pomocou informácií z týchto paketov sa rozhodne o type spojenia, teda type párovacej metódy ktoré boli popísané v odseku 2.9.

Pri rôznych párovacích metódach je rozdiel v bezpečnosti. Je potrebné vhodne interpretovať mieru zabezpečenia zariadenia. Pre túto potrebu bol vytvorený takzvaný **bezpečnostný vektor** (`security level`). Tento bezpečnostný vektor udáva mieru zabezpečenia zariadenia. Môžeme si ho predstaviť ako bodovací systém so stupnicou 0 až 10, pričom 0 je najmenej bezpečný typ spojenia a 10 je najviac zabezpečený typ spojenia. Do verzie Bluetooth 4.2 sa dajú všetky párovacie metódy zaradiť to týchto 11 stupňov / bodov. Laikovi toto hodnotenie stačí napríklad pri výbere z viacerých zariadení. Pokročilý používateľ si môže vypísať o jednotlivých vektoroch podrobnosti pomocou prepínača. Ten obsahuje informácie o type spojenia a potenciálnych hrozbách. Príklad takéhoto výpisu je vidieť na obrázku 5.3.

```
|----- Security Level  1: ---------------------------------|
|   - LE Legacy Pairing - Passkey Entry                     |
|   - responder displays, initiator inputs                  |
|   - authenticated                                         |
|   - never protects against passive attacks                |
|   - should protects against active MITM attacks           |
|   - this type of the connection is NOT secure             |
-------------------------------------------------------------
```

Obr. 5.3: Príklad informácii o bezpečnostnom vektore

Pre detektor boli navrhnuté kľúčové fukncionality pre užívateľa:

- **detekcia** - samotná detekcia bezpečnostných vlastností zariadenia

- **zobrazenie výsledkov detekcie** - zobrazenie výsledkov z detekcie zariadení pomocou prehľadnej tabuľky
- **odstránenie záznamu** - možnosť odstránenia záznamu detekovaného zariadenia
- **podrobnosti o bezpečnostnom vektore** - možnosť vypísania krátkej informácie o bezpečnostnom vektore
- **podrobnejší výpis** - možnosť prepnutia prepínačom do pokročilého módu, v ktorom je vypísaných viac informácií

Zobrazenie výsledkov detekcie obsahuje ID zariadenia, MAC adresu, bezpečnostný vektor, prípadne názov zariadenia (ak zariadenie typu `slave` túto informáciu odošle). Pri podrobnejšom výpise obsahuje okrem vyššie uvedených informácií ešte detailnejšie informácie, ako je verzia Bluetooth, počet zachytených párovacích paketov či použitie šifrovania. Pri detekcii je možné prehľadne zobraziť pomocou prepínača zachytené pakety. Pokročilejší používateľ tak bude môcť sledovať proces odchytu.

Nástroju Ubertooth sa nie vždy podarí odchytiť `connect` paket a následne získať všetky informácie, ktoré sú potrebné k analýze dát. Niekedy sa stane, že sa odchytí len jeden z dvoch párovacích paketov. Preto bol navrhnutý systém, ktorý by mal takéto výpadky minimalizovať. Napríklad, ak sa odchytí len polovica dát, uložia sa, aj keď sú nekompletné. Ak sa pri následnej detekcii odchytí druhá polovica dát a prvá bude chýbať, použije sa už prvá odchytená uložená polovica dát pre analýzu spojenia. Chýbajúce dáta sa teda nahradia už zachytenými dátami. Informácie o zariadeniach sa ukladajú do súborov, a tak je ich možné prenášať, či uchovávať. Spojovaním (`mergovaním`) je teda možné celú detekciu urobiť spoľahlivejšou. Bohužiaľ, úspešnosť nástroja Ubertooth sa pri odchyte nedá nijako inak ovplyvniť. Týmto spôsobom sa ale detekcia značne urýchlila. Používateľ môže zariadenia mazať zo zoznamu zariadení. Môže zmazať jedno zariadenie, ktoré identifikuje pomocou ID, alebo všetky zariadenia.

Tento návrh detekčného nástroja by mal poskytnúť jednoduchú obsluhu a jednoduchý výpis aj pre menej znalého užívateľa. Snaží sa minimalizovať nedostatky zariadenia Ubertooth, ktorého odchyt paketov nie je bezchybný.

5.3 Monitorovací nástroj

Monitorovací nástroj má za úlohu pasívne monitorovať okolie a hlásiť prípadné nežiadúce spojenia. Keďže je v súčastnej dobe je na trhu veľa nezabezpečených zariadení, je veľká pravdepodobnosť, že používateľ bude vlastniť aspoň jedno slabo zabezpečené zariadenie. Preto klasický prípad použitia monitorovacieho nástroja je, že používateľ má svoje Bluetooth Low Energy zariadenia v dosahu nástroja Ubertooth. Následne spustí monitor, ktorý ho bude informovať / varovať, ak sa niekto pokúsi na tieto zariadenia pripojiť, prípadne ak sa jeho zariadenia pokúsia pripojiť na cudzie zariadenia. Systém mu dá informáciu, kto sa na jeho zariadenie pripojil. Útočníkove zariadenie si navyše môže používateľ uložiť. Nabudúce, ak sa útočníkové zariadenie priblíži, bude používateľ varovaný.

Z hľadiska funkčnosti je Ubertooth nastavený na odpočúvanie jednoho z troch advertising kanálov. Sleduje úkony zariadení a vyhodnocuje pakety. Dôležitým paketom je `connect` paket, ktorý sa používa na spojenie zariadení. Ak sa takýto packet objaví v súvislosti s užívateľovým zariadením, mal by byť o tom informovaný.

Pre potreby monitoringu boli navrhnuté 3 zoznamy zariadení. Prvým zoznamom je takzvaný **whitelist**. To sú zariadenia ktorých komunikácia je povolená. Druhým zoznamom zariadení je **unknown** zariadenia. To sú neznáme zariadenia, ktoré sa pohybujú v okolí. Nové

zariadenia sa automaticky pridávajú do tohto zoznamu. Poslednou kategóriou / zoznamom je **blacklist**. Toto sú zariadenia, ktoré užívateľ označil ako nebezpečné. Zariadenia v zoznamoch spravuje používateľ. Nové zariadenia sa pridávajú do zoznamu `unknown` automaticky. Automaticky sa rozpozná, či ide o zariadenie typu `master` alebo `slave`. Ak zariadenie typu `slave` odošle svoje meno, bude tiež pridané do zoznamu (zariadenia typu `Master` svoje meno neposielajú). Vygeneruje sa mu ID a zaradí sa do tabuľky zariadení. Následne používateľ môže toto zariadenie, ktoré bude identifikovať podľa ID, preradiť do iného zoznamu. V zozname používateľ takisto vidí MAC adresu zariadenia.

Samozrejme je nevyhnutné, aby používateľ vedel o aktivite zariadení v okolí. Preto bola navrhnutá takzvaná **história** aktivity zariadení. Každá aktivita zariadení na `advertising` kanáli sa ukladá. Či už je to `advertising indication` paket alebo `scan request` paket. Aktivitu si užívateľ môže zobraziť. Pri zobrazení histórie uvidí informáciu o dátume a čase, id zariadenia a skratku zoznamu do ktorého patrí, názov zariadenia (ak bol niekedy predtým zachytený), aktivitu a cieľové zariadenie.

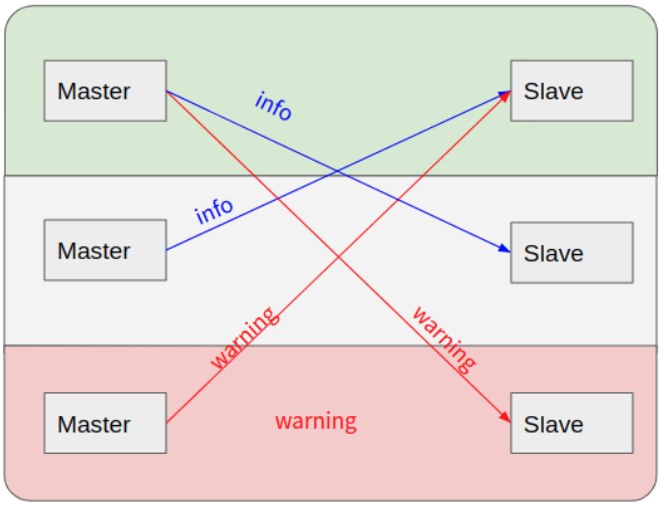
Samozrejme celková história na advertising kanáli môže obsahovať veľa záznamov. Preto bol navrhnutý prepínač, pomocou ktorého si vieme nájsť históriu, ktorá sa týka aktivity zariadenia so zadaním ID. Takto je možné si pozrieť históriu jedného zariadenia a dohľadať tak zariadenia, ktoré sa naňho pripájali. Hoci toto riešenie značne zjednodušilo zobrazenie histórie, aj tak obsahovalo príliš veľa záznamov, a to konkrétne `advertising indication` paketov. Tie zariadenia typu `slave` vysielajú, ak nie sú pripojené k žiadnemu zariadeniu typu `master`. Tieto pakety zariadenie vysiela približne raz za sekundu, čo pri odpojení zariadenia na hodinu spôsobilo množstvo záznamov. Preto sa tieto záznamy agregujú a vypisuje sa už len začiatok a prerušenie / koniec posielania `advertising indication` paketu. Začiatok je prvý advertising paket po dlhšom čase. S ukončením vysielania je to komplikovanejšie. `Connect request` (požiadavka na spojenie) a `Scan` paket sú dve akcie, ktoré sa považujú za legitímne prerušenie odosielania. Ak sa však ani jeden z týchto paketov sa nepodarí odchytiť a zariadenie dlhšie ako 15 sekúnd nepošle `advertising indication` paket, označí sa za stratené. „Stratiť“ sa zariadenie môže, ak sa napríklad prenesie mimo dosah Ubertooth zariadenia alebo ak sa naňho pripojí zariadenie typu `master`, ktoré nie je v dosahu Ubertooth zariadenia. Predposledným prípadom je, ak sa spojí so zariadením typu `master` na inom advertising kanáli. Bluetooth Low Energy má 3 advertising kanály. Ubertooth umožňuje odchytávať iba jeden v danom momente. Pre sledovanie všetkých advertising kanálov by sme potrebovali 3 Ubertooth zariadenia. To by minimalizovalo tento problém. Posledný prípad je, ak Ubertooth neodchytí pakety napríklad z dôvodu zarušenia kanálu.

Užívateľ si vie zobraziť históriu zariadenia a zariadenie pridať do zoznamu. Poslednou vecou, ktorú bolo potrebné navrhnúť sa stal systém **upozornení** (`warning / info`). Upozornenia používateľa informujú o nežiadúcich úkonoch. **Info** sa zobrazí ak:

- zariadenie vo `whiteliste` typu `slave` je spojované so zariadením typu `master` v zozname `unknown`
- zariadenie vo `whiteliste` typu `master` sa spája so zariadením typu `slave` v zozname `unknown`

Warning sa zobrazí ak:

- zariadenie vo `whiteliste` typu `slave` je spojované so zariadením typu `master` v zozname `blacklist`
- zariadenie vo `whiteliste` typu `master` sa spája so zariadením typu `slave` v zozname `blacklist`



Obr. 5.4: Jednoduchá schéma fungovania upozornení

- zariadenie v zozname `blacklist` sa nachádza v okolí

Znázornenie tohto procesu zobrazovania `warning / info` sa nachádza na obrázku 5.4. Podrobnejší popis fungovania `warning / info` funkcie je popísané v sekcii 6.2.

Používateľovi je umožnené mazanie histórie a zariadení. Konkrétne môže vymazať zariadenie a históriu, všetky zariadenia a históriu, alebo len históriu. Mazanie zariadení bez zmazania histórie nie je možné, pretože ak by sme vymazali zariadenie, ktoré sa nachádza v histórii, pri jej zobrazení by nebolo možné vypísať korektne aktivity spojené s chýbajúcim zariadením. Ak by sme vymazali viac zariadení, nebolo by z histórie jasné, ktoré zariadenia spolu komunikovali. V najhoršom prípade, pri vymazaní všetkých zariadení, by história ukazovala interakciu „chýbajúcich zariadení“ s „chýbajúcimi“ zariadeniami. Takéto zobrazenie nie je žiadúce, a preto nie je užívateľovi umožnené.

Návrh na fukncionalitu obsahuje tieto položky:

- **monitoring** - monitoring okolia, zobrazovanie upozornení na nežiadúce akcie
- **zobrazenie celej histórie** - chronologické zobrazenie celkovej aktvivity na advertising kanáli
- **zobrazenie histórie jedného zariadenia** - možnosť zobrazenia histórie aktivity jedného zariadenia, ktoré identifikujeme jeho ID
- **3 zoznamy zariadení** - zoznamy `whitelist / unknown / blacklist`
- **Automatické pridanie nového zariadenia** - zistenie typu zariadenia, vygenerovanie nového nepoužitého ID, uloženie do zoznamu `unknown` spolu s jeho MAC adresou
- **preradenie zariadenia do zoznamu** - preraďovanie zariadenia medzi 3 zoznamami
- **odstraňovanie** - odstraňovanie zariadení a histórie

Tento návrh by mal poskytnúť užívateľovi jednoduchú a prehľadnú manipuláciu s nástrojom. Nástroj užívateľ na začiatku nakonfiguruje, následne ho ponechá spustený bez potreby úkonov potrebných k jeho funkčnosti.

5.4 Zhrnutie návrhu

Cieľom návrhu detekčného a monitorovacieho nástroja bolo vytvoriť dva nástroje, ktoré sa budú navzájom svojou funkcionalitou dopĺňať. Prvý z nich, detekčný nástroj, pomôže otestovať Bluetooth Low Energy zariadenia. Bude používaný na začiatku. Druhý nástroj, monitorovací, bude používaný na monitorovanie zariadení a ich akcií v okolí a bude používateľa informovať o nežiadúcich aktvitách. Monitorovanie bude používané dlhodobo. Spojená funkcionalita bola zavrhnutá, pretože pri detekcii nemožno vykonávať plnohodnotný monitoring.

Pri návrhu bolo uvažované o dvoch typoch používateľov. Prvým je neznalý používateľ, teda taký, ktorý sa v danej problematike nevyzná. Tento typ používateľa ocení jednoduché výpisy a bodový systém bezpečnosti zariadenia. Druhým typom používateľa je pokročilý užívateľ, teda užívateľ znalý problematiky. Ten ocení podrobné výpisy zo zachytávania packetov ako aj podrobnejšie výpisy informácií o zariadeniach. Pri monitorovaní bola vynaložená veľká pozornosť prehľadnosti histórie jedného zariadenia.

V návrhu detekčného a monitorovacieho nástroja bolo potrebné brať do úvahy nedokonalosť odchytávania nástroja Ubertooth. V prípade detektora je to nedokonalé sledovanie spojenia a časté výpadky v jeho sledovaní. Pri snahe minimalizovať počet odchytov pre úspešnú detekciu zariadenia bol navrhnutý systém, pri ktorom sa ukladajú aj neúplné dáta. Pri monitorovacom nástroji je to počet `advertising` kanálov. Na odstránenie tohto problému by sme však potrebovali 3 zariadenia Ubertooth.

Kapitola 6

Implementácia detekčného a monitorovacieho nástroja

V tejto kapitole sa pokúsim priblížiť implementáciu detekčného a monitorovacieho nástroja. Obsahuje prehľad zaujímavých častí riešení pri práci, ako aj popis a riešenie niektorých problémov. Pred čítaním tejto časti odporúčam prečítať predchádzajúcu kapitolu, ktorá hovorí o návrhu (časť 5). Pri implementácii bola veľká snaha o použitie len potrebného, a teda čo najmenšieho počtu knižníc. Tak isto bol kladený dôraz na rýchlosť behu programu. Je potrebné, aby program zaberal čo najmenej miesta a potreboval čo najmenej výkonu, pretože sa čakáva použitie nástroja aj na bránach a minipočítačoch s nižším výkonom.

Pri implementácii bol zvolený **programovací jazyk** C. Je síce náročnejší na implementáciu, ale poskytuje prístup do úrovní blízkych hardvéru. Ďalšou veľkou výhodou je rýchlosť behu programu v tomto jazyku, čo je potrebné pri veľmi malých časových intervaloch medzi paketmi. Pri implementácii bolo potrebné využiť **knižnice** pre zariadenie Ubertooth. Tie majú ale závislosť ešte na ďalších knižniciach. A to na `libusb` a `libbtbb`. Knižnica `libusb` sa využíva na komunikáciu s USB hardvérom. Teda na komunikáciu s Ubertooth zariadením. Knižnicu `libbtbb` používa Ubertooth tools balíček na dekódovanie Bluetooth paketov. Následne `Ubertooth lib` sa používa na správne nakonfigurovanie a odchyt Bluetooth Low Energy packetov. Okrem týchto knižníc sú potrebné už len štandardné knižnice jazyka C. Tie sa používajú napríklad pre prácu s časom, či na prácu s reťazcami. Používateľ nainštaluje nástroj pomocou vytvoreného CMAKE súboru. Implementácia obsahuje prehľadné komentáre typu javadoc.

Aj keď detekčný a monitorovací modul sú dva samostatne funkčné programy, bol vytvorený hlavičkový súbor `MainStructures.h`, ktorý je spoločný pre oba programy. Ten obsahuje štruktúry, ktoré sú rovnaké a spoločné pre oba programy. Obsahuje štruktúry, ktoré sa používajú pri dekódovaní komunikačného protokolu Bluetooth Low Energy. Štruktúry pre všetky potrebné pakety, ktoré bolo potrebné dekódovať v rámci tejto bakalárskej práce. Štruktúra pre každý paket obsahuje názov paketu a položky, ktoré majú presnú veľkosť v bitoch. Tak je umožnené ľahké mapovanie štruktúr na dáta pri ich dekódovaní. Príklad takejto štruktúry je znázornený na obrázku 6.1. Tieto štruktúry paketov boli vytvorené na základe oficiálnej dokumentácie Bluetooth [3]. Tak isto bol pre lepšie pochopenie paketov veľmi nápomocný program Wireshark, pomocou ktorého bolo odkontrolované správne porozumenie dokumentácie. Tak isto bol vhodný pre odkontrolovanie správnej interpretácie Bluetooth Low Energy paketov. Tieto štruktúry paketov následne značne zjednodušili implementáciu oboch programov. Hlavičkový súbor `MainStructures.h` ďalej obsahuje hlavné

```
struct packet_adv_scan_req_t {
    uint32_t access_adress; //(connection address)

    uint8_t pdu_type:4;
    uint8_t rfup:2; //rfu in byte with pdu
    uint8_t txadd:1;
    uint8_t rxadd:1;

    uint8_t length:6;
    uint8_t rful:2; //rfu in byte with length

    uint8_t scan_address[6];
    uint8_t adv_address[6];
} ; //SCAN_REQ
```

Obr. 6.1: Príklad štruktúry paketu ktorá sa používa na mapovanie dát

štruktúry pre Monitor a Detektor. A to štruktúru pre ukladanie informácií o spojení, štruktúru pre monitorované zariadenie a štruktúru pre záznam v histórii. Bližšie informácie o týchto štruktúrach sú v ďalších častiach.

6.1 Popis implementácie detekčného nástroja

Hlavnou úlohou detekčného nástroja je dekódovanie paketov Bluetooth Low Energy, ich analýza a následná interpretácia výsledku. Pre jeho použitie je potrebné pripojiť zariadenie typu `master` a `slave` po prvýkrát v jeho blízkosti. Ak by zariadenia boli pripojené už predtým, nedošlo by k výmene párovacieho paketu. Ak zariadenia boli pripojené už v minulosti, je tu možnosť na zariadení typu `master` využiť funkcie na „zabudnutie“ zariadenia.

Súbory implementované v rámci tvorby detektora a ich stručný popis:

- `BLEDetector.c/.h` - kontrolovanie argumentov a spúšťanie funkcií
- `BLEDetectorFunc.c/.h` - konfigurácia nástroja Ubertooth a dekódovanie paketov
- `BLEDevices.c/.h` - ukladanie informácií o komunikácii so zariadeniami a práca so súbormi
- `BLEDetectorPrint.c/.h` - analýza a interpretácia výsledkov detekcie na `stdout`
- `AddressList.c/.h` - ukladanie MAC adries zariadení s ich menami pre príslušné uloženie
- `MainStructures.h` - štruktúry paketov a základná štruktúra pre ukladanie informácií o spojení

Hlavným „main“ súborom je `BLEDetector`, ktorý má za úlohu **spracovanie argumentov** programu pomocou funkcie `getopt`. Sleduje správne použitie argumentov, obsahuje funkciu na vypísanie bezpečnostných vektorov do `stdout`. Ak sú argumenty programu správne, volá príslušné funkcie.

Funkcie pre konfigurovanie a dekódovanie paketov sa nachádzajú v súbore `BLEDetectorFunc`. Za pomoci funkcie `DetectorStart` sa nakonfiguruje zariadenie Ubertooth pre odchyt Bluetooth Low Energy paketov. A to na počúvanie 37. `advertising` kanálu. V prípade zachytenia `connect request` paketu nasleduje spojenie na dátových kanáloch. Získané dáta poskytne funkcii `DetectorReceiver`. Teda funkcii, ktorá sa zavolá pri zachytení dát. Tej sa prostredníctvom parametera funkcie odovzdajú získané dáta. Táto funkcia

je pomerne obsiahla. Funguje ako veľký konečný deterministický stavový automat. Dáta získáva pomocou ukazovateľa, takže je ľahko možné ich namapovať na štruktúru. Na začiatku sa namapujú na štruktúru pod názvom `packet`. Následne sa určí, či ide o `advertising` paket, alebo o `dátový` paket. Ostatné pakety sa zahodia, a to z toho dôvodu, že Ubertooth okrem valídnych paketov posiela plno „šumu“. Tento šum bol prítomný aj za použitia zariadenia Ubertooth v odstienenej komore. Pri valídnych „nešumových“ paketoch sa číta bit po bite a posúva sa medzi stavmi, pričom každý stav si mapuje štruktúru paketu. Keď sa dostaneme do posledného stavu a namapuje sa presná štruktúra istého typu paketu, zistia sa z paketu potrebné informácie. Tie sa ukladajú do štruktúry `DeviceConnectionInfo`. Tá obsahuje všetky hodnoty, ktoré je možné o zariadení získať a sú potrebné pre analýzu bezpečnosti. Následne sa táto štruktúra uloží do binárneho súboru `devices.dat`. Ak sa v tomto súbore už nachádza záznam o odpočúvanom zariadení, hodnoty sa pridajú k už zachyteným hodnotám. V prípade paketu `ADV_IND` sa tento paket uloží ešte aj do lineárne viazaného zoznamu. Druhýkrát zachytený `ADV_IND` paket od toho istého zariadenia sa ignoruje a nevypisuje sa na `stdout`.

V prípade `CONNECT_REQ` paketu začne Ubertooth sledovať dátové kanály, na ktorých sa pohybuje spojenie zariadení a zachytávať pakety obsahujúce informácie o verzii Bluetooth, či použitia šifrovania. Aj tieto informácie sa ukladajú do štruktúry `DeviceConnectionInfo` a následne do súboru `devices.dat`. V prípade, ak sa podarí odchytiť aj dvojicu paketov `PAIRING_REQ` a `PAIRING_RSP`, obsahuje súbor všetky potrebné dáta pre analýzu bezpečnosti spojenia.

Často sa stane, že zariadenie Ubertooth „stratí“ prebiehajúce spojenie, alebo nezachytí všetky pakety. V tomto prípade sa môže stať, že napríklad informácie z paketu `PAIRING_REQ` budú chýbať. Ak sa pri druhom odpočúvaní spojenia daného zariadenia podarí zachytiť `PAIRING_RSP`, chýbajúce informácie sa doplnia a bude možné vykonať analýzu aj bez zachytenia `PAIRING_REQ` paketu. A to je možné vďaka funkcii v súbore `BLEDevices`, ktorá spája chýbajúce dáta. Ak sa ten istý typ dát odchytí dvakrát, uložia sa novšie dáta.

Tento súbor ďalej obsahuje funkcie pre prácu s uloženými zariadeniami. Umožňuje ich mazať a zobrazovať užívateľovi. Zariadenia sú identifikované pomocou MAC adresy a v prípade úkonov používateľa sa identifikujú za pomoci unikátneho ID zariadenia. Je to štvor miestne číslo, ktoré sa vygeneruje zariadeniu, ak je zachytené po prvýkrát.

Pri zobrazení bezpečnostných informácií zariadenia užívateľom sa vykoná analýza získaných dát. Tá obsahuje štyri kroky, ktoré slúžia na rozhodnutie o verzii párovania. Táto verzia priamo súvisí s úrovňou zabezpečenia. Tieto kroky sú popísané podrobne v časti 2.9. Výsledkom analýzy je bezpečnostný vektor.

Používateľ si vie zobrazovať základné, alebo podrobnejšie informácie o zariadeniach. Tie si vyberie pomocou prepínača. Pre zorientovanie sa vo funkcionalite programu Bol napísaný stručný help a README.

6.2 Popis implementácie monitorovacieho nástroja

Hlavnou úlohou monitorovacieho nástroja je dekódovanie paketov, ukladanie zariadení do zoznamu, ukladanie histórie akcií na `advertising` kanáli a varovanie užívateľa pred nežiadúcimi akciami zariadení. Pri jeho používaní si používateľ označí svoje zariadenia, ako zariadenia patriace do zoznamu `Whitelist`. Ak prebehne pripojenie iného zariadenia k jeho zariadeniu, dostane užívateľ upozornenie. Súbory implementované v rámci tvorby monitorovacieho nástroja a ich stručný popis:

- `BLEMonitor.c/.h` - kontrolovanie argumentov a spúšťanie funkcií
- `BLEMonitorFunc.c/.h` - konfigurácia nástroja Ubertooth a dekódovanie paketov
- `BLEMonitorDeviceList.c/.h` - ukladanie informácií o zariadení do súboru, zoznamy (whitelist/unknown/blacklist)
- `BLEMonitorHistory.c/.h` - ukladanie histórie akcií na `advertising` kanáli do súboru
- `BLEMonitorPrint.c/.h` - výstup z histórie, záznamov o zariadeniach a warning/info správ na `stdout`
- `AddressList.c/.h` - ukladanie MAC adries zariadení s ich menami pre príslušné uloženie
- `MainStructures.h` - štruktúry paketov a základná štruktúra pre ukladanie informácií o spojení

Hlavná „`main`" funkcia sa nachádza v súbore `BLEMonitor`. Má za úlohu kontrolu správnosti argumentov. V prípade ich správnosti spúšta konkrétne funkcie. Pre spustenie funkcie monitorovania sa použije funkcia `MonitorStart` zo súboru `BLEMonitorFunc`. Táto funkcia nakonfiguruje a spustí zariadenie Ubertooth. A to v režime, kedy pasívne odpočúva na `advertising` kanáli. Narozdiel od detektora, v prípade `CONNECT_REQ` paketu, zostane odchytávať naďalej advertising kanáli. V prípade zachytenia dát, pošle dáta funkcii `MonitorReceiver`, ktorej ich odovzdá v rámci argumentu funkcie ako ukazateľ do pamäte. Na tieto dáta následne mapuje štruktúry z `MainStructures.h` a dekóduje informácie. Tento proces mapovania dát je veľmi podobný detekčnému nástroju, no líši sa akciami pri dekódovaní rôznych paketov. V prípade monitorovania sú podstatné štyri pakety. Prvým je `ADV_IND` paket, ktorým nespárované zariadenie indikuje svoju prítomnosť v okolí. Tento paket sa uloží do lineárne viazaného zoznamu a v prípade voľby výpisu paketov pomocou prepínača sa už druhýkrát nevypisuje. Druhúm je `SCAN_REQ`, ktorým zariadenie typu master zisťuje informácie o zariadení typu slave. Naopak paket `SCAN_RSP` je odpoveď na túto požiadavku. Zariadenie v ňom, okrem iného, oznamuje svoje meno. To sa uloží do lineárne viazaného zoznamu pre ďalšie použitie. Posledným paketom je `CONNECT_REQ`, pomocou ktorého sa zariadenia pripájajú. V komunikácii budú ďalej pokračovať na `dátových` kanáloch.

V prípade, ak je zariadenie zachytené po prvý raz, je potrebné uložiť ho do **zoznamu zariadení**. Je nepodstatné, pri ktorom zo štyroch dôležitých paketov, spomínaných vyššie, bolo zachytené. Pre jeho neskoršiu identifikáciu sa mu vygeneruje unikátny štvormiestny identifikátor a zariadenie následne sa uloží do binárneho súboru `monitordevices.dat`. Uloží sa jeho adresa, ID a informácia, či sa ide o zariadenie typu `slave` alebo `master` a informácia o zaradení do zoznamu „`unknown`", kde sa nachádzajú „neznáme" zariadenia. Ak bola zachytená, alebo sa v budúcnosti zachytí aj informácia o mene, bude doplnená. Používateľ si toto zariadenie môže označiť za „svoje", respektíve za dôveryhodné / povolené zariadenie. Vykoná to použitím prepínača a ID zariadenia. V tomto prípade sa zo súboru vyhľadá dané zariadenie a zmení sa jeho zaradenie do zoznamu `whitelist`. Ak chce užívateľ označiť zariadenie ako potencionálne nebezpečné, zaradí ho do zoznamu `blacklist`.

Ak má používateľ zaradené zariadenia do zoznamov, bude dostávať **upozornenia o vzniknutých akciách**. Ide o dva typy upozornení. Typ `warning` a typ `info`. Medzi týmito typmi sa rozhoduje na základe závažnosti akcie. Môže nastať šesť typov upozornení, a to:

1. Info: Slave zo zoznamu `whitelist` je pripojovaný k neznámemu zariadeniu zo zoznamu `unknown`
2. Info: Master zo zoznamu `whitelist` sa pripája k neznámemu zariadeniu zo zoznamu `unknown`

3. Warning: Slave zo zoznamu `blacklist` je v okolí
4. Warning: Master zo zoznamu `blacklist` je v okolí
5. !!!Warning: Slave zo zoznamu `blacklist` je pripojovaný k zariadeniu zo zoznamu `whitelist`!!!
6. !!!Warning: Master zo zoznamu `blacklist` sa pripája k zariadeniu zo zoznamu `whitelist`!!!

O type upozornenia rozhoduje funkcia `MonitorDeviceCheckForWarningInfo`, ktorá je volaná pri každom odchyte zariadenia. Skontroluje zoznam zariadení a vykoná nasledovné upozornenia: pri pakete `ADV_IND` a `SCAN_RSP` sa generuje upozornenie č.3. Pri zvyšných dvoch paketoch je situácia zložitejšia. Pri pakete `SCAN_REQ` sa generuje upozornenie č.3 a č.4, a to podľa situácie popísanej tabuľkou 6.1.

		Master		
		whitelist	unknown	blacklist
Slave	whitelist			č.4
	unknown			č.4
	blacklist	č.3	č.3	č.3,4

Tabuľka 6.1: Tabuľka o výbere typu upozornenia pri pakete `SCAN_REQ`

V prípade paketu `CONNECT_REQ` sa generuje upozornenie popísané tabuľkou 6.2. V nej je možné vidieť, že pripájania zariadení z rovnakých zoznamov sa nehlásia. Ak sa pripája zariadenie zo zoznamu `blacklist` so zariadením zo zoznamu `blacklist` alebo `unknown`, je používateľ varovaný o prítomnosti zariadenia zo zoznamu `blacklist` v okolí. Ak sa pripájajú zariadenia zo zoznamu `whitelist` so zariadeniami zo zoznamu `unknown`, je na túto skutočnosť užívateľ iba upozornený.

		Master		
		whitelist	unknown	blacklist
Slave	whitelist		č.1	č.6
	unknown	č.2		č.4
	blacklist	č.5	č.3	č.3,4

Tabuľka 6.2: Tabuľka o výbere typu upozornenia pri pakete `CONNECT_REQ`

V prípade odchytu paketu sa tento paket uloží do **histórie**. Funkcie pre potreby histórie sa nachádzajú v súbore `BLEMonitorHistory.c/.h`. Do binárneho súboru `deviceslog.dat` sa uloží typ paketu, zistí sa ID zariadenia typu `master` a typu `slave` a uloží sa čas akcie. V prípade vypísania celej histórie akcií na advertising kanáli sa vypíšu všetky záznamy. Pre prehľadnejší výpis o jednom zariadení bola implementovaná funkcia `MonitorHistoryShowOne`, ktorej ako parameter vložíme ID zariadenia, určeného na zobrazenie. Uvedená funkcia vyhľadá záznamy len s príslušným ID zariadenia. Tieto záznamy obsahujú veľa záznamov o `ADV_IND` paketoch. Preto sa v tejto funkcii agregujú, t. j. vypíše sa len prvý `ADV_IND` paket, ktorý je braný ako štart vysielania týchto paketov. Ak nepríde do 15 sekúnd ďalší, označí sa spojenie so zariadením ako stratené. Ak príde `CONNECT_REQ` alebo `skenovací` paket, považuje sa to za legitímne prerušenie vysielania `ADV_IND` paketov. Používateľ si tak vie

zobraziť prehľadnú históriu jedného zariadenia. Pri mazaní zariadení je vždy potrebné zmazať históriu. Ak by sme nezmazali zariadenia spolu s históriou, nebolo by možné dohľadať informácie o zariadení a spôsobovalo by to nekonzistenciu dát.

Kapitola 7

Experimentovanie, testovanie a vzorové výstupy

Táto kapitola je venovaná popisu experimentov a testovania detekčného a monitorovacieho nástroja vytvoreného v rámci bakalárskej práce. Takisto obsahuje aj vzorové výstupy jednotlivých nástrojov. Obsahuje aj popis problémov, ktoré sa vyskytli pri testovaní detekčného a monitorovacieho nástroja a nebolo ich možné vyriešiť. Väčšina z nich vyplýva z obmedzení nástroja Ubertooth a vlastností Bluetooth Low Energy.

Testovanie prebiehalo dvoma spôsobmi. Prvým spôsobom bolo testovanie bez použitia nástroja Ubertooth, ktoré malo overiť funkcionalitu monitoru a detektora bez limitácií, ktoré má zariadenie Ubertooth. Boli vytvorené dáta simulujúce výstup dát, ktoré by normálne pochádzali zo zariadenia Ubertooth. Simulovanými dátami boli postupne overené všetky funkcionality. Pri preverovaní jednotlivých funkcionalít a možných scenárov bola potvrdená funkcionalita samostatného monitorovacieho a detekčného nástroja.

Druhým spôsobom bolo testovanie za použitia zariadenia Ubertooth, teda testovanie monitorovacieho a detekčného nástroja s reálnymi dátami zachytenými pomocou nástroja Ubertooth. Pri tomto testovaní bol použitý Android Smartphone OnePlus 6 ako zariadenie typu `master` a ako zariadenie typu `slave` Fitbit Flex 2. Podľa návrhu bolo zariadenie Ubertooth pripojené do počítača. Na počítači bol spustený monitorovací alebo detekčný modul, pričom v okolí zariadenia Ubertooth boli so zariadeniami vykonávané rôzne operácie. Napríklad pripojenie, pohyb zariadenia po okolí, vyhľadávanie zariadení, párovanie, či strata zariadenia z okolia. Následne boli získavané výstupy z nástrojov. Pri testovaní celku sa objavili viaceré problémy, ktoré sú spolu s možnými vylepšeniami do budúcnosti popísané v časti 7.3.

7.1 Výsledky testovania detekčného nástroja a výstupy detekcie

Úlohou detekčného nástoroja je detekovať bezpečnostné problémy v Bluetooth Low Energy komunikácii. V návrhu je popísaná situácia použitia, kedy pred zapnutým detekčným nástrojom pripojíme (po prvýkrát) zariadenia a necháme ich prípadne spárovať. Je preto nutné, aby používateľ mal svoje zariadenie typu `master` a typu `slave`. To je výhodné, pretože zariadenie typu `master` môže disponovať staršou verziou Bluetooth a preto nemôže využiť novšie metódy párovania (pridané vo verzii 4.2), ktoré sú bezpečnejšie. Z tohto dô-

vodu je vhodné, aby sa zariadenia testovali s cieľovým zariadením typu `master` (teda so zariadení, ku ktorému má používateľ v úmysle zariadenia typu `slave` pripájať).

Aj napriek tomu bolo uvažované o pridaní ďalšej funkcionality, ktorá by bola klasicky použitá pri výbere medzi dvomi zariadeniami. A to takej, že zariadenia by sa pripájali pomocou zabudovaného Bluetooth zariadenia, ktoré by obsahoval počítač, prípadne brána so zapojeným zariadením Ubertooth. To znamená, že po spustení detekčného nástroja by sa pomocou vstavaného Bluetooth zariadenia vyhľadali nespárované zariadenia typu `slave`. Následne by sme sa pomocou tohto zabudovaného Bluetooth postupne na nich pripájali a snažili sa ich párovať. A to do vtedy, kým by sme neodchytili všetky potrebné pakety. Bohužiaľ, pri skúšaní pripájania sa na Bluetooth Low Energy zariadenia vstavaným Bluetooth neboli zariadením Ubertooth žiadne pakety odchytené. Preto bolo od pridania tejto ďalšej funkcionality upustené.

Detekčný nástroj je možné spustiť v dvoch režimoch. V základnom, kde nie je real-time výpis o zachytených paketoch a v druhom móde, určenom pre pokročilejších používateľov. V tomto móde si môže užívateľ pomocou prepínača zobraziť zachytené pakety a tak pozorovať priebeh zachytávania. Príklad výpisu pri zachytávaní paketov je na výpise 7.1.

```
$ BLE-Detector -d -p
Detecting devices with printing
==================BTLE Detector=====Started: 16-04-2019 13:30:53==========================
Date:      Time:    Type:     Access address:  Data:                          (S - M)
================================================================================================
16-04-2019 13:30:53 ADV_IND          8e89bed6 | S: 40:16:3b:0b:7c:50 (public) -> Broadcast
16-04-2019 13:31:03 SCAN_RSP         8e89bed6 | S: 8e:8b:fe:59:33:4a (public) -> Broadcast
16-04-2019 13:31:21 SCAN_REQ         8e89bed6 | S: d0:5b:b8:53:6b:92 (public) <- M: 50:46:1f:3c:ef:98
16-04-2019 13:31:21 SCAN_RSP         8e89bed6 | S: d0:5f:b8:53:6b:92 (public) -> Broadcast
                                              |                      Name: BeeWi SmartClim
16-04-2019 13:31:52 CONNECT_REQ      8e89bed6 | S: d8:e9:ee:b9:ee:a3 (random) <- M: 50:46:1f:3c:ef:98
                                              |                      New address: af9aa428
16-04-2019 13:31:52 LL_VERSION_IND   af9aa428 | Version of Bt: 7 - 4.1
16-04-2019 13:31:52 LL_VERSION_IND   af9aa428 | Version of Bt: 7 - 4.1
16-04-2019 13:31:52 LL_FEATURE_REQ   af9aa428 |
16-04-2019 13:31:53 SMP_PAIRING_REQ  af9aa428 | IO Cap: 0x04 KeyboardDisplay
                                              | OOB: 0x00 Data not present
                                              | MITM: 1
```

Výpis 7.1: Príklad vypisovania paketov pri ich zachytávaní

Vo výpise 7.1 je možné vidieť, že vypísané záznamy obsahujú dátum a čas, adresu spojenia, adresu zariadenia (typu `slave` vľavo a typu `master` vpravo). V prípade, ak paket obsahuje ďalšie informácie, sú informácie dekódované a pridané do záznamu. V uvedenom výpise je možné vidieť pripojenie zariadenia s odchytením jedného párovacieho paketu.

Ak sa nám podarí zachytiť pakety potrebné pre analýzu dát, zobrazí sa v tabuľke odchytených zariadení bezpečnostný vektor zariadenia. Túto tabuľku je možné takisto spustiť v zjednodušenom režime (výpis 7.2) alebo v pokročilejšom režime (výpis 7.3).

```
$ BLE-Detector -s
Showing list of devices
======================Device Info=================================
|  ID: |      Address:     | Sec. Vector: |   Name:              |
| 8228 | d8:e9:ee:b9:ee:a3 | 0-Unsecure   | Flex 2               |
| 9934 | 9e:b5:03:07:01:12 |  No pairing  |                      |
| 3819 | 66:00:00:00:00:00 | 9-Secure     | AAB                  |
```

Výpis 7.2: Tabuľka zariadení v zjednodušenom režime

```
Printing out devices detailed report:
======================Device Info=================================================================
|  ID: |      Address:     |   Name:              | Version: | Enc: | Pair: | Sec. Vector: |
```

```
| 8228 | d8:e9:ee:b9:ee:a3 | Flex 2              | 4.1(2/2) |  No  | (2/2) | 0-Unsecure   |
| 9934 | 9e:b5:03:07:01:12 |                     | N/A(0/2) |  No  | (0/2) |  No pairing  |
| 3819 | 66:07:00:b5:e9:ee | AAB                 | 4.1(2/2) |  Yes | (2/2) | 9-Secure     |
```

Výpis 7.3: Tabuľka zariadení v pokročilejšom režime

V detailnejšom zobrazení možno vidieť aj počet párovacích paketov a počet paketov označujúcich verziu Bluetooth. Záznamy zariadení je možné mazať.

Pri testovaní bola overená funkčnosť detekčného nástroja. Či už na simulovaných dátach, alebo na dátach zo zariadenia Ubertooth. Tak isto bola overená funkčnosť spájania informácií z paketov, t. j. ak sa nezachytia dáta na prvý pokus a zachytí sa len časť dát a druhýkrát táto časť bude chýbať, je možné previesť analýzu komunikácie a interpretovať bezpečnostný vektor. Často krát sa stalo, že zariadenie ubertooth nezaregistruje pripojenie alebo stratí zachytené pripojenie. Bohužiaľ, túto limitáciu prístroja sa nepodarilo obísť. Presnejší popis problémov spolu s možnými riešeniami je popísaný v časti 7.3.

7.2 Výsledky testovania monitorovacieho nástroja a výstupy monitorovania

Monitorovací nástroj má za úlohu sledovať `advertising` kanál, oznamovať nežiadúce akcie a zariadenia v okolí. Podľa návrhu sa na začiatku očakáva, že používateľ na chvíľu tento detekčný nástroj spustí a ten odhalí zariadenia v jeho okolí. Následne si zariadenia zaradí do zoznamov (whitelist / blacklist / unknown). Funkcionalita odchytu zariadení, ich pridania do zoznamu (unknown) a možnosti menenia zoznamu bola overená a je funkčná. Príklad výpisu zoznamu zariadení je možné vidieť na výpise 7.4. Tak isto bolo overené mazanie zariadení, jedného z nich alebo celého zoznamu.

```
$ BLE-Monitor -d
Showing list of devices
====================== Devices ========================================
|  ID: |      Address:     | Role:  | Type:     |   Name:              |
| 4209 | 3c:7b:c1:16:94:0f | master |  unknown  |                      |
| 1942 | 5a:31:5a:85:89:ac | slave  |  unknown  |                      |
| 6630 | fe:4d:0e:d5:03:73 | master | whitelist |                      |
| 2006 | a3:ee:b9:ee:e9:d8 | slave  | blacklist | Flex 2               |
```

Výpis 7.4: Zoznam zariadení

Počas monitoringu je obdobne možné vypisovať pakety pri ich zachytení. Štýl výpisu je totožný s výpisom pri detekčnom nástroji 7.1. Samozrejme, tento výpis neobsahuje pakety z dátových kanálov. Ďalej bola otestovaná funkčnosť histórie zariadení na advertising kanáli. Bola overovaná spolu s výpisom histórie jedného zariadenia. Bolo možné otestovať všetky možné výstupy, keďže počas testovania monitoringu boli zachytené všetky možné situácie (pripojenie, odpojenie, zariadenie v okolí, strata zariadenia). Príklad výpisu pri zobrazovaní histórie je možné vidieť na výpise 7.5. Spolu s históriou zariadení bolo otestované jej mazanie.

```
$ BLE-Monitor -s -o 2006
Showing history of one device:
========= History of device id: 2006 Name:Flex 2  Role: Slave ======
|    Date & Time:     |  Action:
| 11-04-2019 22:09:33 | ADV START    ----to--->  Broadcast
                      |
                      |
| 11-04-2019 22:09:53 | SCAN REQ     <--from---  8194(W)
| 11-04-2019 22:09:53 | SCAN RSP     ----to--->  Broadcast
```

```
| 11-04-2019 22:09:57 | SCAN REQ     <--from---  6874(U)
| 11-04-2019 22:10:05 | CONNECT REQ  <--from---  8194(W)
| 11-04-2019 22:12:23 | ADV START    ----to--->  Broadcast
              |
              |
| 11-04-2019 22:12:44 | Device LOST! Long time between packets!
| 11-04-2019 22:13:27 | ADV START    ----to--->  Broadcast
              |
```

Výpis 7.5: História jedného zariadenia

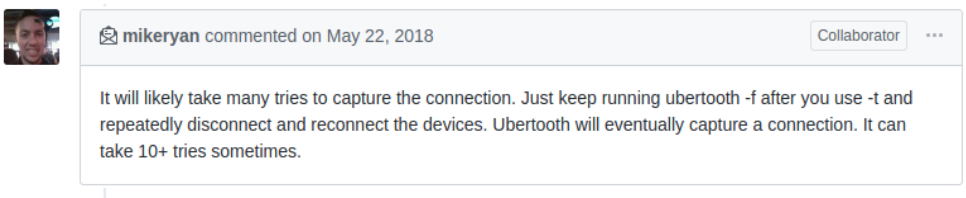
Následne bola overená funkčnosť upozornení. Upozornenie typu info a typu warning sa zobrazovalo podľa návrhu a bolo označené za funkčné.

Bola overená funkčnosť monitorovacieho nástroja, a to na simulovaných aj na reálnych dátach. Občas sa stane, že zariadenie Ubertooth nezachytí paket, čo v tomto prípade nemuselo byť chybou zariadenia Ubertooth, ale stalo sa tak v dôsledku, že Bluetooth Low Energy obsahuje tri `advertising` kanály. Zariadenie Ubertooth vie v jednom čase počúvať iba jeden kanál a môže sa stať, že napríklad paket `CONNECT_REQ` bol poslaný na inom kanáli.

7.3 Zhrnutie výsledkov testovania a možné vylepšenia

Pri testovaní detekčného a monitorovacieho nástroja bola overená správna funkčnosť nástrojov. Taktiež sa vyskytli problémy, ktoré sa nepodarilo odstrániť. Tieto problémy sú dané obmedzeniami nástroja Ubertooth a vlastnosťami Bluetooth Low Energy.

Prvý problém, ktorý sa vyskytoval pri testovaní detekčného nástroja, bola strata spojenia so zariadeniami. Akonáhle bol zaznamenaný paket `CONNECT_REQ`, zariadenie Ubertooth sa presunulo spolu so zariadeniami na dátové kanály a začalo vykonávať s nimi `frequency hopping`. Zariadenie Ubertooth zachytilo pár paketov, no následne komunikáciu stratilo a vrátilo sa na `advertising` kanál. Ďalším podobným problémom bolo, že ak zariadenie Ubertooth následovalo a odchytávalo spojenie na dátových kanáloch, občas nezachytí (vynechá) paket. Ukladaním všetkých dát a ich následným spájaním v prípade opakovaného pokusu boli následky straty paketov čo najviac minimalizované. Boli hľadané riešenia ako tento problém odstrániť. Avšak ich hľadanie bolo neúspešné. Sám vývojár zariadenia Ubertooth sa na svojej projektovej stránke vyjadril, že to niekedy zaberie 10+ pokusov, než sa podarí spojenie zachytiť (možno vidieť na obrázku 7.1).



Obr. 7.1: Vyjadrenie vývojára Ubertooth na otázku úspešného odchytenia spojenia

Druhým veľkým problémom sa stali tri `advertising` kanály. Keďže zariadenie Ubertooth dokáže zachytávať v jednom okamihu iba jeden z nich, je pravdepodobnosť 33,33 %, že paket bude odoslaný (a zachytený) na sledovanom kanáli. V skutočnosti je táto šanca vyššia, pretože pri skúšaní odchytávania paketov na kanáloch 37, 38 a 39 bolo zistené, že si zariadenia najčastejšie posielajú pakety na kanáli 37, na ktorý bolo aj zariadenie Ubertooth nastavené. Tento problém sa týkal detekčného ale hlavne monitorovacieho nástroja. Pri monitorovacom nástroji sa stávalo, že sa zariadenie pripojilo, ale nebol zachytený `CONNECT_REQ`

paket. V tom prípade sa zariadenie označilo za stratené aj napriek legitímnemu ukončeniu vysielania `advertising` paketov. Tento problém by sa dal vyriešiť použitím troch zariadení Ubertooth súčasne, no v rámci bakalárskej práce bolo k dispozícií len jedno toto zariadenie.

Ďalším poznatkom ktorý bol zistený pri odchytávaní paketov bolo, že ani jedno zariadenie do verzie 4.2 nemenilo svoju MAC adresu aj keď ju mali označenú ako náhodnú (random). Správne by mali pri každom spojení túto adresu zmeniť, nestalo sa tak ani raz.

V rámci testovania bola overená správna funkčnosť nástrojov, ale problémy popísané vyššie som nebol schopný riešiť. Nenašiel som spôsob, ako urobiť odchyt Bluetooth Low Energy paketov zariadením Ubertooth spoľahlivejším. Ak zariadenie Ubertooth neposkytne / nezachytí dáta, tak sa nad nimi nemôže urobiť analýza. Až na nespoľahlivý odchyt, však bola implementácia úspešne otestovaná, boli otestované taktiež príslušné funkcionality popísané v návrhu a správnosť implementácie.

Kapitola 8

Záver

Cieľom tejto baklárskej práce bolo vytvoriť nástroj na detekciu bezpečnostných incidentov v Bluetooth Low Energy sieťach. Úlohou bolo vytvoriť nástroj, ktorý bude detekovať bezpečnostné problémy v BLE zariadeniach a tie vhodne intrepretuje používateľovi. Uplatnenie nájde tiež pri zisťovaní miery zabezpečenia zariadenia. Taktiež má nástroj za úlohu sledovať akcie BLE zariadení a v prípade, že zachytí pripájanie nežiadúcich zariadení na zariadenie užívateľa, poskytne užívateľovi o tejto události hlásenie. Všetky vytýčené ciele boli splnené.

V rámci riešenia som sa najprv zoznámil s komunikačným protokolom Bluetooth Low Energy a s možnosťami monitorovacieho nástroja Ubertooth. Následne som sa zaoberal známymi bezpečnostnými incidentmi v BLE sieťach. Zistil som, že väčšina bezpečnostných incidentov je spojená so starými verziami Bluetooth spolu s nevhodným výberom metódy párovania.

Na základe získaných informácií som navrhol detekčný a monitorovací nástroj. Úlohou detekčného nástroja je detekovať bezpečnostné problémy BLE v zariadeniach. Monitorovací nástroj má za úlohu monitorovať akcie BLE zariadení a v prípade nežiadúcich akcií na ne upozorniť používateľa. Oba tieto nástroje využívajú zariadenie Ubertooth One na odchyt BLE paketov. Podarilo sa čiastočne znížiť dopady obmedzenia tohto nástroja. V rámci použitia bolo cielené na užívateľov znalým aj neznalým danej problematiky.

Implementáciu celého systému som realizoval v jazyku C. Vznikli tak dva nástroje, ktoré sa svojou funkcionalitou navzájom dopĺňajú. Pri implemenácii bol braný zreteľ na slabšie parametre zariadení, na ktorých by môhli byť tieto nástroje spustené. Implemenácia bola otestovaná na simulovaných dátach, následne na dátach reálnych, ktoré pochádzali zo zariadenia BLE. Testovaním bolo overené, že sú tieto nástroje funkčné.

Vytvorené nástroje sú vhodné na použitie hlavne v miestach obsahujúcich BLE „smart“ siete. Pri požití týchto nástrojov by som ale do budúcnosti doporučil použitie troch zariadení Ubertooth, pomocou ktorých by sa minimalizovali výpadky odchytených paketov. Dosiahnuté výsledky sa spolu s informáciami, zistenými v mojej bakalárskej práci, použijú v rámci spolupráce so združením CESNET pod záštitou projektu SIoT.[1]

[1] Security Internet of the Things (Zabezpečená brána pro Internet věcí), url: https://www.cesnet.cz/projekty/

Literatúra

[1] Bluetooth: *Bluetooth Pairing Part 1 Pairing Feature Exchange*. Bluetooth Special Interest Group, Marec 2016, [Online; navštíveno 19.01.2019].
URL https://blog.bluetooth.com/bluetooth-pairing-part-1-pairing-feature-exchange

[2] Bluetooth: *Bluetooth Pairing Part 2 Key Generation Methods*. Bluetooth Special Interest Group, Jún 2016, [Online; navštíveno 19.01.2019].
URL https://blog.bluetooth.com/bluetooth-pairing-part-2-key-generation-methods

[3] Bluetooth: *Specification of the Bluetooth System*. Bluetooth Special Interest Group, December 2016, [Online; navštíveno 19.01.2019].
URL
https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=421043

[4] Bluetooth: *Bluetooth Sig - About Us*. Bluetooth Special Interest Group, Január 2019, [Online; navštíveno 19.01.2019].
URL https://www.bluetooth.com/about-us

[5] Bluetooth: *Bluetooth Sig - Connected Device*. Bluetooth Special Interest Group, Január 2019, [Online; navštíveno 19.01.2019].
URL https://www.bluetooth.com/markets/connected-device

[6] Bluetooth: *Bluetooth Sig - Phone, Tablet & PC*. Bluetooth Special Interest Group, Január 2019, [Online; navštíveno 19.01.2019].
URL https://www.bluetooth.com/markets/phone-pc

[7] Bluetooth: *Bluetooth Sig - Smart Home*. Bluetooth Special Interest Group, Január 2019, [Online; navštíveno 19.01.2019].
URL https://www.bluetooth.com/markets/smart-home

[8] Cauquil, D.: *Bluetooth Low Energy Attacks*. Econocom Digital Security, September 2018, [Online; navštíveno 19.01.2019].
URL https://nis-summer-school.enisa.europa.eu/cources/IOT/nis-summer-school-damien-cauquil-BLE-workshop.pdf

[9] Jasek, S.: *GATTACKING'BLUETOOTH SMART DEVICE*. SecuRing, November 2017, [Online; navštíveno 19.01.2019].
URL http://gattack.io/whitepaper.pdf

[10] Lonzetta, A. M.; Cope, P.; Campbell, J.; aj.: *Security Vulnerabilities in Bluetooth Technology as Used in IoT. Journal of Sensor and Actuator Networks*, Júl 2018, doi:10.3390/jsan7030028.

[11] Microchip: *Bluetooth® Low Energy Connection Process*. Microchip Technology, Inc., Január 2019, [Online; navštíveno 19.01.2019].
URL http://microchipdeveloper.com/wireless:ble-link-layer-connections

[12] Microchip: *Bluetooth® Low Energy Discovery Process*. Microchip Technology, Inc., Január 2019, [Online; navštíveno 19.01.2019].
URL http://microchipdeveloper.com/wireless:ble-link-layer-discovery

[13] Microchip: *Bluetooth® Low Energy Physical Layer*. Microchip Technology, Inc., Január 2019, [Online; navštíveno 19.01.2019].
URL http://microchipdeveloper.com/wireless:ble-phy-layer

[14] Microchip: *Bluetooth® Low Energy Security Modes and Procedures*. Microchip Technology, Inc., Január 2019, [Online; navštíveno 19.01.2019].
URL http://microchipdeveloper.com/wireless:ble-gap-security#toc4

[15] MS, J. A.: *Understanding Bluetooth Advertising Packets*. Jún 2014, [Online; navštíveno 19.01.2019].
URL http:
//j2abro.blogspot.com/2014/06/understanding-bluetooth-advertising.html

[16] Padgette, J.; Bahr, J.; Batra, M.; aj.: *Guide to BluetoothSecurity. NIST Special Publication*, ročník 800-121, č. 2, Máj 2017,
doi:https://doi.org/10.6028/NIST.SP.800-121r2.

[17] Ryan, M.: *Ubertooth One*. Január 2011, [Online; navštíveno 19.01.2019].
URL http://ubertooth.sourceforge.net/hardware/one/

Príloha A

Popis „IO Cap“ hodnoty v párovacom pakete

„IO Cap“ indikuje vstupno-výstupné rozhrania zariadenia. Vstupné rozhranie je klávesnica (musí obsahovať možnosť zadania kláves 0-9, tlačidlo na potvrdenie a aspoň dve klávesy, pomocou ktorých je možné indikovať Áno/Nie). Ďalšou možnosťou je Áno/Nie vstup (teda dvojica tlačidiel, pomocou ktorých je možné odpovedať Áno/Nie). Poslednou možnosťou je, že zariadenie nemá ani vstup a ani výstup. Výstupné rozhrania majú dve možnosti. Prvá je numerický výstup (teda zariadenie vie zobraziť 6 miestny kód). Druhou je, že zariadenie nedisponuje výstupným zariadením (ktoré je schopné zobraziť tento kód). Pomocou týchto kombinácií sa vytvorí hodnota prenesená v IO Cap:

- **0x00** - *DisplayOnly* - zariadenie obsahuje výstupné zariadenie schopné zobraziť 6-miestny kód, ale neobsahuje vstupné zariadenie
- **0x01** - *DisplayYesNo* - zariadenie obsahuje výstupné zariadenie schopné zobraziť 6-miestny kód a dvojicu tlačidiel Áno/Nie
- **0x02** - *KeyboardOnly* - zariadenie neobsahuje zariadenie, obsahuje len klávesnicu
- **0x03** - *NoInputNoOutput* - zariadenie neobsahuje žiadne vstupno-výstupné zariadenie
- **0x04** - *KeyboardDisplay* - zariadenie obsahuje výstupné zariadenie schopné zobraziť 6-miestny kód a klávesnicu

Príloha B

Krok 4 Procesu párovania

	Initiator				
Responder	**DisplayOnly**	**Display YesNo**	**Keyboard Only**	**NoInput NoOutput**	**Keyboard Display**
Display Only	Just Works Unauthenticated	Just Works Unauthenticated	Passkey Entry: responder displays, initiator inputs Authenticated	Just Works Unauthenticated	Passkey Entry: responder displays, initiator inputs Authenticated
Display YesNo	Just Works Unauthenticated	Just Works (For LE Legacy Pairing) Unauthenticated Numeric Comparison (For LE Secure Connections) Authenticated	Passkey Entry: responder displays, initiator inputs Authenticated	Just Works Unauthenticated	Passkey Entry (For LE Legacy Pairing): responder displays, initiator inputs Authenticated Numeric Comparison (For LE Secure Connections) Authenticated
Keyboard Only	Passkey Entry: initiator displays, responder inputs Authenticated	Passkey Entry: initiator displays, responder inputs Authenticated	Passkey Entry: initiator and responder inputs Authenticated	Just Works Unauthenticated	Passkey Entry: initiator displays, responder inputs Authenticated
NoInput NoOutput	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated
Keyboard Display	Passkey Entry: initiator displays, responder inputs Authenticated	Passkey Entry (For LE Legacy Pairing): initiator displays, responder inputs Authenticated Numeric Comparison (For LE Secure Connections) Authenticated	Passkey Entry: responder displays, initiator inputs Authenticated	Just Works Unauthenticated	Passkey Entry (For LE Legacy Pairing): initiator displays, responder inputs Authenticated Numeric Comparison (For LE Secure Connections) Authenticated

Obr. B.1: Krok 4 pri výbere párovacej metódy[2]