VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

BEZPEČNOSTNÍ TESTOVÁNÍ ZAŘÍZENÍ S BLUETOOTH
BLUETOOTH DEVICE SECURITY TESTING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Jan Hlaváček**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Petr Dzurenda**
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Jan Hlaváček ***ID:*** 146004
Ročník: 2 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Bezpečnostní testování zařízení s Bluetooth

POKYNY PRO VYPRACOVÁNÍ:

1) Seznamte se s Bluetooth specifikací a se zabezpečením Bluetooth komunikačního kanálu.
2) Vytvořte přehled bezpečnostních rizik technologie Bluetooth.
3) Sestavte vhodný adaptér pro bezpečnostní testování zařízení s Bluetooth, např. Ubertooth one.
4) Pro zvolené testované zařízení s Bluetooth vytvořte přehled možných bezpečnostních rizik a hrozeb.
5) Navrhněte testovací postupy. Zvolte vhodný testovací software a hardware.
6) Zvolené zařízení pomocí těchto postupů otestujte. Výsledky zaznamenejte do testovací zprávy.

DOPORUČENÁ LITERATURA:

[1] WRIGHT, J. – CACHE, J.: Hacking Exposed Wireless, Third Edition: Wireless Security Secrets & Solutions. ISBN 978-0071827638, McGraw-Hill Education, 3 edition, 2015

[2] NITS: Guide to Bluetooth Security, NIST Special Publication 800-121. Rev. 1, http://csrc.nist.gov/publications/nistpubs/800-121-rev1/sp800-121_rev1.pdf

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 24.5.2017

Vedoucí práce: Ing. Petr Dzurenda
Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Cílem práce je rozbor a soupis bezpečnostních rizik technologie Bluetooth, sestavení Bluetooth adaptéru a návrh a sestavení testovacích postupů, které pomohou vyhodnotit bezpečnost testovaného zařízení.

KLÍČOVÁ SLOVA

Bluetooth, penetrační testy, DOS, Bluetooth Low Energy, bezpečnost párování, Bezpečnost, Rizika, Ubertooth

ABSTRACT

The aim of the thesis is analysis and inventory security risks of Bluetooth technology, assembly Bluetooth adapter and proposal and poposal of testing procedures, which will helps evaluate security of tested device.

KEYWORDS

Bluetooth, pentests, DOS, Bluetooth Low Energy, security of pairing, Security, Risks, Ubertooth

HLAVÁČEK, Jan *Bezpečnostní testování zařízení s Bluetooth*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 60 s. Vedoucí práce byl Ing. Petr Dzurenda

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Bezpečnostní testování zařízení s Bluetooth“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Technicka 12, CZ-61600 Brno, Czechia

http://www.six.feec.vutbr.cz

Výzkum popsaný v této diplomové práci byl realizovaný v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Petru Dzurendovi a především konzultantovi panu Ing. Radomíru Svobodovi PhD. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

TEORETICKÝ ÚVOD

0.1 Seznámení s Bluetooth specifikací

0.1.1 Aplikace Bluetooth v praxi

Technologie Bluetooth je určená pro propojení dvou nebo několika (maximálně 8) zařízení v rámci „pico sítě“ (tzv. piconet). V dnešní době se především setkáváme s nasazením v mobilních telefonech, handsfree, chytrých hodinkách a dalších osobních zařízeních. V průmyslovém prostředí se může jednat o různé čtečky, detektory, nebo sondy. V rámci často zmiňovaného pojmu „internet věcí“ se můžeme setkávát s použitím i v zařízeních typu chytrých žárovek, v ledničkách, termostatech atd.

Nejčastěji se užívá zapojení point to point, neboli tzv. „Ad Hoc“, kdy se propojují pouze dvě zařízení. Můžeme se ale setkávát se zapojením point to multipoint, kdy se k jedné „master“ jednotce připojuje několik „slave“ zařízení. Poslední možností je propojení několika point to multipoint sítí, čímž vzniká tzv. Scatter Net.

0.1.2 Princip přístupu k fyzickému médiu Bluetooth 1.0

Bluetooth pracuje v pásmu 2,4GHz. Užití tohoto pásma má své výhody i nevýhody. Mezi výhody patří dobré vlastnosti při šíření v běžném prostředí a především bezlicenční politika tohoto pásma. Hlavní nevýhodou je rušení. Setkáváme se zde s rušícími prvky, ať už ze strany WiFi sítí 802.11b/g/n standardu, tak i ze strany dalších spotřebičů, např. mikrovlnných trub. Aby se u Bluetooth předešlo těmto nežádoucím rušením, byly zvoleny principy, které samy o sobě málo ovlivňují jiné technologie ve stejném pásmu a zároveň samy nejsou tolik rušeny.

Pro komunikaci na fyzické vrstvě se používá tzv. kmitočtové skákání nosné. To znamená, že nosná frekvence se celkem 1600 krát za sekund změní v rámci kmitočtového pásma 2,4GHz. Pro modulaci na každou nosnou se používá šířka pásma 1MHz. Následné kódování jednotlivých symbolů probíhá pomocí modulace GFSK (pomocí Gaussovského filtru kmitočtově omezená fázová modulace). Frekvenční odchylka pro jednotlivé symboly je v rozmezí 140-175kHz. Logická jednička reprezentuje kladnou kmitočtovou odchylku a logická nula reprezentuje zápornou kmitočtovou odchylku.

Nosná frekvence se mění podle předem stanoveného pseudonáhodného schématu, který se vytvoří z adresy zařízení a hodinového taktu řídícího (master) zařízení.

Celkové spektrum Bluetooth je v rozmezí 2400MHz - 2483,5MHz. V tomto frekvenčním pásmu je tedy dostupných 79 rádiových kanálů. Data se přenášejí v paketech v krátkých časových intervalech (TS - Time Slots) pomocí funkce TDD (Time Division Duplex).

Při přenosu dat platí, že doba vysílání na jedné nosné je 625μs. Každý rádiový kanál je rozdělen na timesloty o délce právě 625μs. Timesloty jsou číslovány podle hodinového taktu řídící jednotky. Jeden cyklus má celkem 2×10^{28} timeslotů. Takt řídící jednotky je $312,5\mu$s.

Jednotka master vysílá pouze v každém sudém timeslotu a jednotka slave vysílá jen v lichém timeslotu. Každý paket bývá přenášen na jedné nosné frekvenci.

Timeslot	0	1	2	3	4
Master	vysílání	naslouchání	vysílání	naslouchání	vysílání
Slave	naslouchání	vysílání	naslouchání	vysílání	naslouchání

Obr. 1: Rozdělení vysílací doby master a slave zařízení v Bluetooth.

Během přenosu dat může také dojít k situaci, kdy je třeba odeslat paket, který je delší než jeden timeslot. Potom se vysílání prodlouží na dobu 3 až 5 timeslotů a po odeslání paketu se obnoví původní průběh vysílání. V tomto případě je vysílání paketu provedeno pouze na jedné nosné.

Timeslot	0	1	2	3	4
Master	vysílání	vysílání	vysílání	naslouchání	vysílání
Slave	naslouchání	naslouchání	naslouchání	vysílání	naslouchání

Obr. 2: Obsazení timeslotů při odesílání dlouhého paketu.

Dalším opatřením, kterým Bluetooth snižuje zarušení ISM pásma, je vhodná volba výkonnostní třídy zařízení. Celkem jsou tři třídy:

Tab. 1: Výkonnostní kategorie Bluetooth

Výkonová třída	Maximální výstupní výkon	Nominální výstupní výkon	Minimální výstupní výkon
1	100mW (20dBm)	-	1mW (0dBm)
2	2,5mW (4dBm)	1mW (0dBm)	-
3	1mW (0dBm)	-	-

Při běžném používání se dosah pohybuje cca do 10m. V ideálním prostředí bez překážek je možné používat Bluetooth až na 100m. Ovšem při použití vhodné smě-

rové antény je zdokumentován pokus, kdy byla komunikace Bluetooth odposlouchávána až na vzdálenost přibližně 1,6km.

0.1.3 Struktura paketu

Na začátku paketu se nachází přístupový kód o délce 72 bitů. Ten je dán kombinací MAC adresy daného zařízení a hodinového taktu master zařízení. Přístupový kód je unikátní pro každou síť piconet a kromě synchronizace slouží i k autorizovanému přístupu do sítě.

Obr. 3: Rozdělení vysílací doby master a slave zařízení v Bluetooth.

MAC adresa zařízení je rozdělena na tři části: LAP (24 bitů - přidělena výrobcem), UAP (8 bitů) a NAP (16 bitů). UAP a NAP jsou identifikátorem výrobce.

V záhlaví je přenášeno dohromady 18 informačních bitů, které zabezpečuje kanálové kódování FEC K = 1/3 a délka celého pole je tedy 54 bitů. 3 bity v záhlaví obsahují adresu řízení přístupu na médium, 4 bity nesou informaci o typu paketu, a 3 bity nesou řídící bit. Na konci záhlaví je jeden bit pro ARQ a bit pro kontrolu chyb v záhlaví.

Celková délka paketu se může pohybovat v rozmezí od 126 do 2871 bitů.

Přenos dat může probíhat v synchronním módu. Ten se využívá pro hovorové signály a komunikace probíhá rychlostí 64kbit/s v každém směru.

V asynchronním módu přenosu datových signálů může být přenos asymetrický, v jednom směru maximální přenosová rychlost 723,2kbit/s a 57,6 kbit/s ve směru druhém, nebo může být přenos symetrický, s obousměrnou rychlostí až 433,9 kbit/s.

0.1.4 Sestavení spojení

Aby se vytvořilo spojení mezi dvěma zařízeními, musíme jedním zařízením vyhledávat zařízení ostatní. Pro tuto akci spustíme proceduru nazvanou Inguiry. Po její aktivaci začne zařízení měnit frekvenci nosné dvakrát rychleji, tedy 3200 krát za sekundu. Celkem se prochází 32 kanálů podle zvolené posloupnosti. Vyhledávající zařízení vždy vysílá na dvou po sobě jdoucích frekvencích a poté čeká na odezvu. Jelikož vyhledávající zařízení vysílá rychleji než hledané, je zaručeno, že se brzy potkají vysíláním a naslouchánim na stejné frekvenci. Procedura Inguiry může trvat

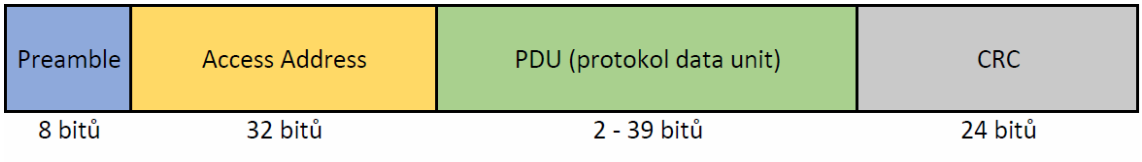
maximálně 10,24 sekund. Průměrná doba kompletního sestavení spojení je mezi 4-6 sekundami.

0.2 Bluetooth Low Energy (BLE)

Se standardem Bluetooth 4.0 přišel na scénu pojem Low Energy. Nová norma Bluetooth se zaměřuje na přenos malých objemů dat s výraznou energetickou úsporou oproti předešlým standardům.

0.2.1 Struktura paketu BLE

Již samotný rámec BLE je odlišný:



Obr. 4: Struktura paketu Bluetooth Low Energy standardu

Každý paket začíná 8 bitovou preambulí a poté následuje 32 bitů adresy. Jedná se o stejný unikátní identifikátor jako v klasickém Bluetooth standardu. Za tímto blokem adresy následuje variabilní blok dat v rozmezí od 2 do 39 bitů. A na konci paketu je umístěn 24 bitový kontrolní součet. V porovnání s klasickým Bluetooth paketem se velikost režijních dat na jeden paket zmenšila na poloviční hodnotu a zároveň došlo ke zkrácení datové části. To má za následek zkrácení doby vysílání a tedy i úsporu energie.

BLE také zavádí dva nové protokoly. Namísto RFCOMM protokolu se využívá Attribute Protocol a na místo Serial Port profilu se začal používat tzv. General Attribute Protocol.

0.2.2 Využití BLE

Bluetooth Low Energy standard nachází využití především v senzorických zařízeních, jednak v průmyslovém prostředí, jednak v medicínských přístrojích. Z toho důvodu je dnes kladen vysoký nárok na bezpečnost a spolehlivost.

Dalším odvětvím využití jsou dnes tzv. „gadgety“. Prvním typickým představitelem jsou např. chytré hodinky. Jako další mohou být programovatelné žárovky, různé displeje, ovladače, nebo i „chytré ledničky“.

0.2.3 Sestavení spojení BLE

Při použití standardu Low Energy probíhá sestavení spojení s určitými obměnami. Pro zahájení komunikace se využívá tří kanálů, konkrétně 2402MHz, 2426MHz a 2480MHz. Kanály se používají jak pro vyhledání zařízení, tak pro začátek spojení. Tyto kanály nazýváme Advertising channels. Jsou rozmístěny v různých částech spektra, aby se předešlo rušení ze strany sítí 802.11.

BLE zařízení se přepíná mezi několika operačními módy:

- Advertising - vysílání Advertising paketů, aby bylo zařízení vyhledatelné a aby druhé zařízení bylo schopné zjistit, jaké služby nabízí.
- Scanning - naslouchání na předem daných frekvencích a čekání na přijetí Advertising paketů.
- Master - má na starosti sestavení spojení a koordinaci více zařízení v rámci jedné piconet sítě. Může se k němu připojit i více slave zařízení.
- Slave - připojuje se k master zařízení a komunikaci s ostatními zařízeními koordinuje přes master.

0.3 Verze Bluetooth standardu

- Bluetooth 1.0 - (2002)
- Bluetooth 1.2 - (2005), Adaptivní frekvenční skákání - maximální přenosová rychlost 721kbps, modulace nosných frekvencí pomocí GFSK
- Bluetooth 2.0 - (2004), EDR (Enhanced Data Rate) - zvýšena propustnost na 2Mbps, užity modulace $\pi/4$DQPSK a 8DPSK
- Bluetooth 2.1 - (2007), EDR - zvýšena propustnost na 3Mbps, zvýšena bezpečnost při párování
- Bluetooth 3.0 - (2009), HS (High Speed) - přenosová rychlost až 24Mbit za pomocí standardu 802.11
- Bluetooth 4.0 - (2010), LE (Low Energy) - nízká energetická náročnost u zařízení, která nevyžadují přenos velkého objemu dat
- Bluetooth 4.2 - (2014), 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) - příprava konektivity pro velký počet zařízení - „internet věcí“
- Bluetooth 5.0 - (prosinec 2016) - nižší spotřeba, vyšší dosah a vyšší stabilita než u verze 4.2, rychlost přenosu až 50Mbit/s

Každá nová verze Bluetooth standardu se zaměřuje na aktuální a budoucí potřeby komunikujících zařízení. Již od verze 2.0 můžeme sledovat rostoucí navyšování přenosové rychlosti. Bohužel se příliš nehledělo na spotřebu, protože většinou šlo o

krátké přenosy dat mezi notebookem a mobilním telefonem, o synchronizaci kalendáře a kontaktů, nebo o potřebu přenášet zvuk na krátkou vzdálenost.

Verze 4.0 již ale reaguje na vývoj nových zařízení, jako jsou například chytré hodinky, kde je naopak kladen extrémní důraz na minimální spotřebu energie a není nutné přenášet dlouhé datové zprávy.

Velká očekávání jsou právě od nově schválené specifikace Bluetooth 5.0, která spojuje do jednoho standardu jak Low Energy funkce, tak i potřebu IPv6 konektivity a možnost přenosu velkého objemu dat, například streamované HD video.

1 BEZPEČNOST BLUETOOTH

1.1 Základní principy bezpečnosti Bluetooth

Každý komunikační standard musí při svém návrhu počítat s bezpečností už na fyzické vrstvě.

- Při návrhu Bluetooth se jako základní bezpečností prvek proti odposlouchávání komunikace využilo vlastnosti pseudonáhodného přeskakování mezi kmitočty. Proto i když bude útočník naslouchat na jedné nebo několika frekvencích, zachytí jen fragmenty komunikace. Druhou vlastností pseudonáhodného FHSS je to, že bez znalosti PNP není útočník schopen určit, na které frekvenci bude příště vysílající jednotka vysílat.
- Další bezpečnostní vlastností jsou klíče odvozené z PIN kódu. Tento kód může být v některých zařízeních, která mají omezená konfigurační rozhraní, předkonfigurován. V jiných zařízeních (např. mobilní telefon) mohou být voleny uživatelem.
- Pro zajištění autentizace zařízení se používá 128 bitový sdílený klíč.
- Aby byla zajištěna důvěrnost dat, používá se proudová šifra s konfigurovatelnou délkou od 8 do 128 bitů.

1.2 Zabezpečení Bluetooth služeb

Bezpečnost Bluetooth služeb se opírá o tři základní bezpečnostní principy:

- **Autentizace** - komunikující strany vzájemně ověřují svoji totožnost, tedy zda jsou tím, za koho se vydávají.
- **Důvěrnost** - je zajištěno šifrování zprávy, nebo celého komunikačního kanálu, bez znalosti klíče není možné zprávu přečíst.
- **Autorizace** - povolení přístupu uživatele k předem definovaným službám.

Bluetooth specifikace nabízí celkem tři úrovně bezpečnosti pro dané spojení.

1. **Bez zabezpečení** - jedná se o promiskuitní režim. Zařízení umožňuje navázat komunikaci s kterýmkoliv zařízením, které je v dosahu a snaží se navázat spojení. Nejčastěji se s tímto režimem setkáváme u Bluetooth Handsfree sluchátek.
2. **Bezpečnost na úrovni služeb** - zde probíhá autorizace přístupu ke konkrétním službám zařízení.
3. **Bezpečnost na úrovni spoje** - před vlastním navazováním spojení jsou ze strany zařízení inicializovány bezpečnostní kroky autentizace a šifrování komunikačního kanálu.

1.3 Princip sestavení bezpečného spojení

Při zahájení párovacího procesu dvou zařízení se nejprve vygeneruje inicializační klíč. Ten se vytváří na základě shodných PIN klíčů na obou zařízeních, na základě unikátní adresy (BD_ADDR) zařízení, které zahájilo spojení a pomocí ověřovatelem náhodně vygenerovaného čísla, odlišného pro každý přenos.

PIN kód může být dlouhý od 8 do 128 bitů. V praxi se nejčastěji používá 4 nebo 6 číslic. Adresa zařízení má 48 bitů a je veřejná spolu s náhodným číslem, které má délku 128 bitů. Toto náhodné číslo se mění pro každý přenos.

Během inicializační fáze není výměna informací nijak zabezpečena. Na základě inicializačního klíče se generuje klíč spoje. Ten komunikující zařízení sdílejí a na jeho základě probíhá autentizace a šifrování spoje. Tento klíč je tajný a nikdy se nevysílá.

1.4 Bezpečnostní hrozby Bluetooth

1.4.1 Nalezení zařízení

U většiny zařízení je již po prvním spárování nežádoucí, aby jejich existence, nebo vysílání bylo odhaleno. Proto dnešní mobilní telefony nabízejí vysílání svého názvu většinou jen po dobu 2 minut od manuálního zapnutí Bluetooth a po uplynutí této lhůty přestanou svůj název vysílat.

Důležité je, že kromě názvu zařízení, je možné si v tomto čase zjistit, jaké služby jsou na něm spuštěné a jakou má svoji unikátní BD adresu.

Když chceme nalézt nějaké zařízení, tak se vysílají broadcast požadavky na nalezení zařízení. Pokud je funkce „viditelnosti“ v cílovém zařízení zapnutá, potom na zdrojovou adresu odpovídá svojí BD_ADDR.

Zařízení je ale možné objevit i pokud je vypnutá funkce odpovídání na vyhledávací broadcast požadavky. Komunikaci můžeme zahájit přímo na jeho BD_ADDR. Tu můžeme zkusit zjistit pomocí „bruteforce“ metody, tedy postupným zkoušením všech možných MAC adres. Což je proces, který trvá dlouhou dobu. Aby se tato doba vyhledávání zkrátila, snažíme se jako útočník zjistit například výrobce čipu hledaného zařízení, nebo i modelové číslo. Tím pomocí databáze IEEE můžeme určit jaká bude první část BD_ADDR a podstatně tak zkrátit dobu potřebnou pro vyzkoušení všech možných kombinací.

Další možností jak odhalit námi hledané zařízení, je podle jeho názvu. Název velké většiny zařízení je ponechán v defaultním stavu, proto pokud v roli útočníka získáme skenováním názvy, jako jsou například UE6300 a G620S-L01, tak jestliže víme, že typem zařízení oběti je mobilní telefon, potom můžeme vyloučit zařízení

UE6300, protože se jedná o tovární označení konkrétního modelu Televizoru Samsung.

Nalezení zařízení je často klíčovým momentem v útoku, i když samo o sobě se jedná o malou hrozbu. Pokud je vše správně zabezpečeno, tak útočník nemusí být úspěšný, i když zařízení nalezne.
Ochrana: vypnutí odpovídání na broadcast pakety, ideálně u nestacionárních zařízení provádět vyhledávání zařízení v neveřejných prostorách.
Nebezpečí: nízké
Typ zařízení: všechna zařízení

1.4.2 Default PIN

Některá Bluetooth zařízení mají v sobě od výroby předdefinovaný PIN kód pro párování s dalšími zařízeními. Může to být z důvodu omezeného konfiguračního rozhraní nebo chyby výrobce.

Nejčastěji se s tímto problémem můžeme setkat např. u Handsfree sluchátek. Útočník tak může zkusit jen několik kombinací PINů a být úspěšný. Nejčastějšími Default PINy bývají skupiny 4 čísel např: 0000, 1111, 1234, 9999. Pokud ale známe cílové zařízení, jaký je to výrobce nebo model, tak si konkrétní PIN můžeme najít v manuálu, nebo datasheetu produktu.
Ochrana: změna defaultního PINu, omezení vysílacího výkonu, případně využití zařízení pouze v neveřejných prostorách
Nebezpečí: vysoké
Typ zařízení: nezkonfigurovaná zařízení, Handsfree

1.4.3 OBEX

Protokol OBEX byl adoptován do Bluetooth standardu za účelem přenosu souborů. V mnoha zařízeních je ale možné spouštět přenos souborů bez předešlého potvrzení přenosu a tím pádem je otevřen prostor pro stahování dat ze zařízení, nebo pro podsouvání vlastních dat, virů, nebo upravených konfigurací a tak dále.
Ochrana: v případě nevyužívání přenosu souborů - vypnout služby OBEX, zkontrolovat vyžadování potvrzení před zahájením přenosu
Nebezpečí: vysoké
Typ zařízení: především starší mobilní telefony, nebo kapesní počítače

1.4.4 RFCOMM

RFCOMM je bezdrátové sériové rozhraní, které má za účel nahradit komunikaci přes sériový kabel. Pomocí RFCOMM protokolu můžeme s využitím speciálních příkazů

zařízení ovládat a nastavovat.

Problém bývá většinou ze strany výrobce. Protokol nemusí vyžadovat autentizaci. RFCOMM funkcionalita bývá sdělena pomocí SDP protokolu (Service Discovery Protocol). Pokud ji přes SDP není možné nalézt, je i tak možné, že je v zařízení RFCOMM spuštěný. Potom se útočník může pokusit přímo připojit. (Existuje celkem 60 kanálů pro současné použití.)

V případě připojení na RFCOMM terminál je možné například z mobilního telefonu vytáčet hovory na placené linky, nebo odesílat SMS zprávy.

Ochrana: vynucení autentifikace před připojením na RFCOMM, zvážit nutnost spuštění této služby.

Nebezpečí: vysoké

Typ zařízení: většina zařízení Bluetooth do verze 3.0

1.4.5 Chybná implementace protokolu Bluetooth

Bluetooth protokol není triviální a nabízí velké možnosti, jak ho do konkrétního zařízení implementovat. Často se proto stává, že ze strany výrobce vznikne chyba při programování Bluetooth části daného zařízení a otevře tak vrátka útočníkům.

Známou a často opomíjenou implementační chybou bývá ponechání zařízení v módu, kdy odpovídá na všechny „discovery“ požadavky a je tak velmi snadno k nalezení.

Asi nejčastěji se při konkrétních implementacích stává, že zůstanou aktivní i nepotřebné profily. Potom se otevírají dvířka k dalším typům útoků na dané zařízení.

Často také vznikají chyby, které umožňují užívat některé Bluetooth služby zařízení bez autentifikace, nebo dokonce převzít kompletní kontrolu nad komunikací nebo nad zařízením.

Poslední implementační chybou bývá nastavení zbytečně vysokého výkonu pro vysílaní.

Abychom předešli v kritických aplikacích právě těmto chybám ze strany výrobce, je dobré vybírat výrobky renomovaných výrobců s ověřenou servisní podporou. U neznačkových zařízení nemusí být oprava firmwaru nikdy vydána.

Ochrana: pravidelné kontroly a aktualizace firmware vydané výrobcem.

Nebezpečí: střední až vysoké

Typ zařízení: všechna zařízení

1.4.6 Bluetooth viry

Bluetooth viry bývají vždy orientovány na nějaký konkrétní systém nebo zařízení a využívají právě některých známých chyb v implementaci protokolu nebo spoléhají

na uživatelskou hloupost.

Největší rozmach byl v době nasazení operačního systému Symbian. Pokud vir je v zařízení aktivován může například mazat data nebo odesílat citlivé informace. Bluetooth protokol využívá většinou jen pro své šíření. Na pozadí se neustále vyhledávají dostupná zařízení a nabízí se jim odeslání souboru, který obsahuje samotný vir. Protože uživatel musí potvrdit přijetí souboru, snaží se vir dosáhnout přijetí častým opakováním pokusu odeslání, kdy uživatel soubor přijme z důvodu, aby už nebyl dále obtěžován, nebo kvůli lákavému názvu souboru, jako například „tapetazima.jar“, nebo „supergame.jar“.

1.4.7 Odchycení linkového klíče

Pro zachycení linkového klíče je na prvním místě potřebné najít a identifikovat probíhající komunikaci mezi námi hledanými zařízeními.

V druhém kroku musí dojít k identifikaci zařízení, které plní funkci master a identifikaci zařízení které plní funkci slave. Současně se zjišťuje jejich BD_ADDR.

V dalším kroku si útočník změní svoji BD_ADDR na stejnou, jako zařízení s funkcí slave a pošle zařízení s funkcí Master zprávu, že ztratil jeho linkový klíč. Master proto pošle klíč nový, který útočník zachytí.

Tato komunikace včetně linkového klíče je celá zachycena a pomocí aplikace btcracker je možné dekódovat zachycenou komunikaci.
Ochrana: použití dostatečně silného šifrovacího klíče.
Nebezpečí: střední až vysoké
Typ zařízení: všechna zařízení s Bluetooth do verze 3.0

1.4.8 Komunikace na nižší úrovni zabezpečení

Zařízení, která komunikují pomocí „Security Mode 4“ mají povoleno snížit zabezpečení komunikace na verzi nižší, pokud druhé zařízení „Mode 4“ nepodporuje.

V případě útočníka je tedy možné simulovat zařízení, které umí komunikovat pouze v Security Mode 1. Tento mód nevyužívá žádné zabezpečení komunikace a útočník tedy může komunikaci odposlouchávat nebo zaujmout pozici „muže uprostřed“.
Ochrana: umožnit pokles pouze na „Security Mode 3“
Nebezpečí: střední až vysoké
Typ zařízení: Bluetooth od verze 2.1 výše.

1.4.9 Odposlech párování zařízení

V prvních verzích Bluetooth probíhal proces párování nešifrovaně a proto bylo možné po zachycení párovací komunikace kompletně ovládnout ustanovené spojení.

Problém s párováním je znám i u Bluetooth verze 4.0, kdy je možné při zachycení advertisement paketů a úvodní části párování pomocí programu „cracker“ získat linkový klíč a poté odposlouchávat danou komunikaci.
Ochrana: provádět proces párování zařízení v uzavřených neveřejných prostorách.
Nebezpečí: vysoké
Typ zařízení: některé verze Bluetooth standardu.

1.4.10 SSP Just Works - (Secure Simple Pairing)

Tento standard párování navazuje spojení bez zadávání kódu PIN. Konkrétně se s ním můžeme setkat například u bezdrátových sluchátek.

U tohoto protokolu není žádná ochrana proti útoku MITM (Men in the middle) - mužem uprostřed. Z toho důvodu by se tento protokol měl vždy používat jen v aplikacích, které nebudou sloužit k provozu kritických služeb.
Ochrana: nasazení v málo důležitých a nekritických službách
Nebezpečí: vysoké
Typ zařízení: Bluetooth verze 2.1 a 3.0

1.4.11 MITM - Men in the middle

Při útoku typu mužem uprostřed, jak ze svého názvu vypovídá, dokáže útočník získat kontrolu nad spojovacím kanálem komunikujících stran. Tím vzniká prostor nejen pro odposlouchávání kanálu, ale hlavně možnost zasahovat do spojení a posílaných dat.

Pro Bluetoth Low Energy tuto zranitelnost využívá např. aplikace GATTacker, která vytvoří z útočníkova zařízení klon oběti a poté pomocí advertisement zpráv donutí protější stranu komunikaci směrovat právě na něho.
Ochrana: šifrování na linkové vrstvě.
Nebezpečí: velmi vysoké
Typ zařízení: všechna bez šifrovaného spojení.

1.4.12 Zneužití jiného zařízení k útoku

Dnes prakticky každý mobilní telefon obsahuje Bluetooth adaptér a velké množství z nich má i připojení k mobilním datům. Zde vzniká poměrně velké riziko ovládnutí konkrétního telefonu, nebo skupiny telefonů a jejich následné zneužití. Útočník tímto

způsobem může například nasadit škodlivý software do telefonu vysoce postaveného zaměstnance podniku a skrze něj odposlouchávat komunikaci ostatních zařízení v kanceláři.

Riziko tohoto útoku se násobí především s množstvím zařízení, která mohou být potenciálně nakažena škodlivým softwarem.

Aby útočník dosáhl co největší šance, že bude nakaženo zařízení poblíž oběti, může se zaměřit na infikování stránek které jsou danou skupinou lidí navštěvovány, nebo vytvořením zadních vrátek například ve firemní nebo školní mobilní aplikaci.

Pokud dojde k ovládnutí velkého počtu zařízení na relativně malé ploše, může toho být zneužito i pro dálkově řízené DOS útoky. Typicky se může jednat o lokality jako jsou školy, firmy, obchodní centra.

Ochrana: antivirová ochrana mobilního zařízení a kontrola, zda na zařízeních není zbytečně nebo neplánovaně zapnutá funkce Bluetooth

Nebezpečí: střední až vysoké

Typ zařízení: veškerá zařízení s Bluetooth modulem, se kterými lze komunikovat vzdáleně přes jiný protokol, typicky přes internet.

1.4.13 DOS (Denial of Services)

V dnešní době zažívají výrazný rozmach různé formy útoků DOS, tedy útoků na odepření služeb. Principem útoku je snaha o zahlcení daného zařízení nebo skupiny zařízení, takovým způsobem, že buď vytíží možnosti přenosového média natolik, že nemůže probíhat žádná jiná komunikace, nebo dojde k útoku na prostředky daného zařízení, pomocí zasílání velkého množství požadavků a tím pádem k zahlcení procesoru nebo operační paměti a následnému kolapsu systému a nemožnosti obsluhovat regulérní požadavky.

DOS na Bluetooth zařízení může útočník provádět oběma zmíněnými způsoby.

První z nich je zahlcení přenosového média. Je to ideální způsob v případě, kdy je potřeba pomocí Bluetooth technologie přenášet větší objemy dat a pomocí DOS útoku na přenosové médium tak můžeme kriticky zpomalit nebo úplně přerušit danou komunikaci.

Pro zahlcení pásma můžeme využít např. generátor bílého šumu naladěný na konkrétní pásmo s dostatečně velkým výkonem a všesměrovými anténami. Druhou a dostupnější variantou zahlcení pásma je sestavení několika 802.11 spojení s frekvencemi naladěnými tak, aby spolu kanály sousedily a spustily vzájemný přenos souborů, aby došlo k maximálnímu vytížení pásma.

Pro druhý typ útoku na zahlcení prostředků zařízení lze využít velkého množství požadavků na danou službu, přičemž zde platí podobná pravidla jako v IP sítích. Je možné odesílat velké množství broadcast paketů na různá zařízení s podvrženými

zdrojovými BD_ADDR. Tím dochází k zahlcení prostředků oběti a k útočníkovi se již odpovědi nevrací.
Ochrana: efektivní ochrana pro zahlcení rádiového spektra neexistuje, jedinou možností v kritických aplikacích je záložní spojení v jiném pásmu, nebo kabelem.

Proti zahlcení prostředků zařízení je možné se chránit dobře nastavenými pravidly tak, aby zařízení nereagovalo na větší množství požadavků, případně aby ve specifických aplikacích například komunikovalo jen s jediným předem definovaným zařízením.
Nebezpečí: střední až vysoké
Typ zařízení: všechny

1.5 Rizika spojená s konkrétními zařízeními a aplikacemi

Pro správné zabezpečení daného zařízení je třeba si uvědomit, jakým typům rizik je konkrétní zařízení v konkrétní aplikaci vystavováno.

Všechna zařízení se musí potýkat s možnostmi odposlechu komunikace nebo DOS útoku.

Mobilní zařízení - telefony, PDA, notebooky - zde hrozí především odcizení citlivých dat z paměti přístroje nebo dat přenášených Bluetooth kanálem. Může dojít k odcizení kontaktů, podvržení vizitek a zpráv.

Statická zařízení - u statických zařízení a trvalých instalací je palčivým problémem především možnost jednoduché identifikace, například pomocí směrové antény a síly signálu. Po identifikaci a zjištění funkce zařízení je možné zařízení strategicky vyřazovat z provozu pomocí dlouhodobých DOS útoků.

Průmyslové prostředí - čtečky, detektory, sondy - u těchto zařízení je hrozbou především jejich možnost vyřazení z provozu, podvržení datových výstupů, nebo narušení komunikace více zařízení, která vzájemně koordinují svoji činnost.

2 VÝBĚR, SESTAVENÍ A OVĚŘENÍ FUNKČNOSTI MODULU

V poslední části práce jsem se zabýval výběrem a sestavením vhodného Bluetooth adaptéru pro potřeby vytvoření testovacích procedur bezpečnosti jednotlivých zařízení s Bluetooth modulem.

Po nastudování problematiky Bluetooth modulů byl vybrán modul Ubertooth One. Jedná se o programovatelný modul. Je založený na Open-Source kódu.

Kolem Ubertooth One je široká komunita uživatelů a na stránkách projektu Ubertooth je velmi podrobná dokumentace, díky které je možné se seznámit s principem fungování modulu a jeho vlastnostmi.

Jsou zde i materiály pro vlastní sestavení modulu - šablona pro vyleptání čtyřvrstvé PCB desky včetně schémata osazení PCB, seznam všech potřebných součástek a také firmwary a bootloader pro zprovoznění vytvořeného modulu.

Sestavení modulu vyžaduje z důvodu použití mikročipů se spodními vývody alespoň horhovzdušnou pájku, aby bylo možné připájet i vývody pod čipem. Tato první část osazení je nejnáročnější na přesnost a bezchybnost připájení všech kontaktů.

Další osazení součástek je již výrazně jednodušší.

Pro zprovoznění byl zvolen virtuální počítač, aby byla v budoucnosti zaručena jednoduchá přenositelnost vytvořeného systému a případně možnost využití persistent režimu.

Operační systém pro testování byl vybrán Kali Linux. Jedná se o distribuci Linuxu založenou na distribuci Debian. Sdílí tedy stejný balíčkovací systém, ale od začátku je sestrojena pro testování síťových služeb a zařízení.

Poslední verze obsahuje i základní podpůrné soubory a programy určené pro Ubertooth modul. Ty můžeme zjistit z příkazové řádky pomocí napsání „ubertooth-“ a dvojitým poklepání tabelátoru. Konzole nám vypíše všechny programy, které byly vytvořeny pro modul Ubertooth.

```
$ ubertooth-
ubertooth-btle     ubertooth-ego      ubertooth-spescan
ubertooth-debug    ubertooth-follow   ubertooth-spescan-ui
ubertooth-dfu      ubertooth-rx       ubertooth-util
ubertooth-dump     ubertooth-scan
```

Po nahrání bootloaderu a firmwaru je třeba funkčnost modulu otestovat. To provedeme pomocí linuxového programu pro práci s Bluetooth adaptéry. Program „hciconfig“ umožňuje zobrazit BD Address našeho zařízení a také jeho stav - zda je zapnuté, nebo vypnuté.

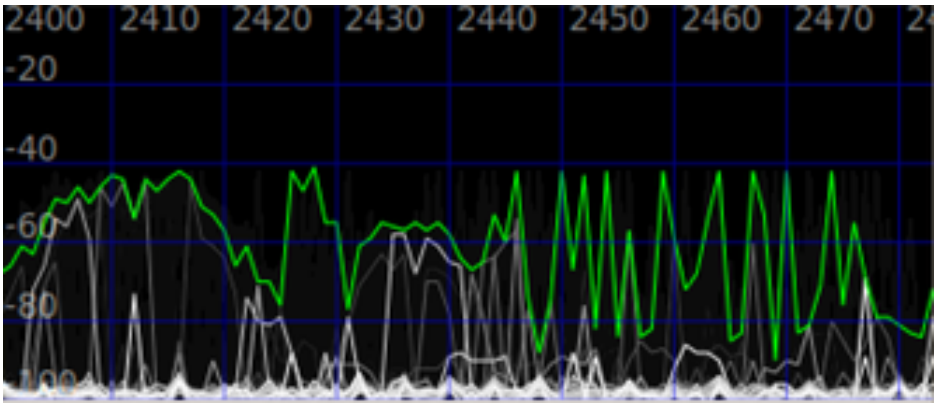
```
$ hciconfig hci0 up
hci0: Type: Primary Bus: USB
      BD Address: 40:2C:F4:C2:C1:7D ACL MTU: 8192:128 SCO
      MTU: 64:128
      UP RUNNING
      RX bytes: 1006 acl:0 sco:0 events:44 errors:0
      TX bytes: 672 acl:0 sco:0 commands:44 errors:0
```

Podle výpisu vidíme, že zařízení běží a komunikuje s operačním systémem. Příkaz „hciconfig“ je dobré používat po každém startu virtuálního počítače a kontrolovat stav zařízení, protože ve většině případů se zařízení přepne do režimu „DOWN“. Tento nedostatek se dá případně odstranit pomocí patřičného skriptu, spouštěného pomocí „crone“ po spuštění systému.

Pro zjištění zda správně funguje i rádiová část modulu je možné spustit skenování spektra. To zajistíme programem, který je vytvořený přímo pro Ubertooth modul.

```
$ ubertooth-spescan-ui
```

Na modulu se rozsvítí diody aktivity rádiového rozhraní a otevře se okno ve kterém se nám začne vykreslovat spektrum v pásmu od 2400MHz do 2483,5MHz. Obr. 2.1



Obr. 2.1: Zobrazení spektra pomocí programu „ubertooth-spescan-ui“

Abychom si na spektru ověřili, že nám ukazuje reálné hodnoty, je možné si na mobilním telefonu spustit funkci „wifi hostspot“, připojit k němu druhé zařízení, započít intenzivní komunikaci (třeba kopírování velkého souboru) a sledovat jak v určité části spektra vzroste jeho obsazení.

3 SESTAVENÍ TESTOVACÍCH PROCEDUR

Předmětem testování je zařízení firmy Honeywell. Za tímto účelem jsem navrhl a provedl několik testovacích postupů. Výsledný testovací protokol není veřejný a slouží pouze pro interní potřeby firmy Honeywell. Vytvořené postupy jsou použitelné i na jiná Bluetooth zařízení a jejich cílem je vytvoření testovacího pracoviště a následné otestování a vyhodnocení bezpečnosti pro nejčastější aplikace.

Každá úloha je koncipována na tři části.

1. Sestavení úlohy, příprava, výběr aplikací
2. Postup testování úlohy
3. Vyhodnocení získaných dat a poznatků

Celkem jsem sestavil a provedl 5 úloh, z nichž první úloha a poslední dvě úlohy jsou aplikovatelné globálně na kterékoliv zařízené vybavené standardem Bluetooth, třetí řeší praktický odposlech a snahu o prolomení Low Energy párovacího procesu a cílem druhé úlohy je pokus o zachycení a prolomení dat šifrovaných na linkové vrstvě u klasického bluetooth.

3.1 Vytvoření testovacího pracoviště

Pro testování jsem zvolil virtuální počítač, do kterého jsem nainstaloval linuxovou distribuci Kali Linux ve verzi 2017.1-amd64. Jako virtualizační prostředí jsem vybral VMware Workstation 12 Player ve verzi 12.5.5 build-5234757.

Pro instalaci je nutné vytvořit virtuální pevný disk o velikosti větší než 12Gb, jinak mohou nastat komplikace už v průběhu instalace operačního systému. Doporučuji tedy zvolit velikost, která bude dostačovat i pro ukládání potřebných dat, instalaci programů a aktualizací.

Na testovací procedury není potřeba extrémní výkon. Bez problémů stačí přiřadit vytvořenému počítači 1-2Gb operační paměti a 1-2 procesorová jádra. Dále je nutné mít k dispozici alespoň dva volné USB porty, ideálně čtyři porty s dostatečným příkonem pro napájení čtyř adaptérů. Konkrétně bude potřeba jeden klasický USB Bluetooth adaptér s podporou verze 4.0 a jeden až tři Ubertooth moduly. Jeden je možné použít pokud testujeme konkrétní zařízení v laboratorních podmínkách a můžeme opakovat proces párování v libovolném množství. Pokud budeme chtít testovat zařízení, které je již v provozu, doporučuji užití tří modulů, aby šlo zachytávat všechny tři advertisement kanály současně.

Při testování se ale často můžeme setkat nutností virtuální počítač ukončit, zvláště pokud je počítač používán i na jinou práci. Problémem je v těchto případech fakt, že úlohy nejde nijak jednoduše uložit. Velkou výhodou potom může

být použití SSD disku pro hostující počítač. Jeho rychlost je znát především pokud si při vypínání zvolíme uložení stavu virtuálního počítače a při dalším zapnutí se stav obnovuje. Dále nedoporučuji ukládání pevného disku virtuálního počítače v rámci lokální sítě, nebo internetu. Odezvy na spuštění programů, nebo samostatného systému se výrazně prodlužují.

Po čisté instalaci operačního systému je velmi vhodné doinstalovat balíček vmware tools, který usnadní přenášení dat mezi virtuálním a hostujícím počítačem. Pro jeho instalaci zvolíme v menu Player – Manage – Install VMware tools. Tím se nám připojí virtuální cd-rom mechanika. Zkopírujeme si soubor VMwareTools-*Verze*.tar.gz do námi zvoleného umístění a provedeme extrahování jeho obsahu. Otevřeme si terminál a přesuneme se do vzniklého adresáře vmware-tools-distrib. Instalaci spustíme pomocí příkazu:

```
./vmware-install.pl
```

Pokud instalujeme rozšíření bezprostředně po čisté instalaci operačního systému, tak by vše mělo proběhnout bez problémů.

Dalším krokem je instalace nástrojů pro práci se zařízeními Ubertooth. To doporučuji provádět podle návodu přímo ze stránek projektu Ubertooth.

https://github.com/greatscottgadgets/ubertooth/wiki/Build-Guide

Instalační manuál udržují autoři aktualizovaný a může se s vydáním nové verze nástrojů nebo firmware lišit. Proto je vhodné postupovat vždy podle aktuálně publikovaného na stránkách výše.

I když jsou nové verze nástrojů a firmwaru vydávány pouze jednou za rok až dva, doporučuji si nainstalovat poslední verzi a v případě, že vše funguje, pracovat s danou instalovanou verzí. Poslední verze je aktuálně z března tohoto roku. Na začátku zpracovávání této práce jsem používal verze aplikací vydané v roce 2015 a v průběhu sestavování úloh jsem si aktualizoval nástroje na poslední verzi. To ale přineslo nutnost aktualizace firmware ve všech Ubertooth modulech a drobné problémy s kompatibilitou, které se mi podařilo vyřešit až opětovnou čistou instalací virtuálního počítače. Z tohoto důvodu doporučuji zůstat u první instalované verze, pokud nová verze nepřinese nové nástroje nebo opravy významných chyb.

V případě, že všechny instalace proběhnou v pořádku, je potřeba otestovat, zda je pracoviště připraveno k použití.

Pokud jsou Ubertooth moduly správně připojeny a detekovány virtuální počítačem, musí svítit LED diody číslo 1V8LED, USBLED a RSTLED. Viz. Obr. 3.1

Připojíme-li modul až po startu virtuální počítače, budeme ho muset zřejmě restartovat a po novém startu systému jej možná i odpojit a připojit. Tento problém se vyskytuje i u klasického Bluetooth modulu. Patrně zde vzniká problém při předávání informací o nově připojeném hardware mezi hostujícím a virtuálním počítačem.

Dále je třeba vždy kontrolovat, zda nám VMware nabízí možnost připojení modulu k virtuálnímu pc. Úspěšné připojení můžeme zkontrolovat pomocí příkazu:

```
hciconfig
```

a musíme dostat podobně vypadající výpis:

```
hci1: Type: Primary  Bus: USB
  BD Address: 00:1A:7D:DA:71:10  ACL MTU: 310:10  SCO
  MTU: 64:8
  DOWN
  RX bytes:574 acl:0 sco:0 events:30 errors:0
  TX bytes:368 acl:0 sco:0 commands:30 errors:0

hci0: Type: Primary  Bus: USB
  BD Address: 00:50:56:E7:B7:7D  ACL MTU: 8192:128  SCO
  MTU: 64:128
  DOWN
  RX bytes:503 acl:0 sco:0 events:22 errors:0
  TX bytes:336 acl:0 sco:0 commands:22 errors:0
```

Systém tak rozpoznává klasický Bluetooth modul jako zařízení hci1 a Ubertooth jako hci0. Důležitý je parametr DOWN, který udává, že zařízení jsou připojená, ale zatím nejsou spuštěná. To provedeme příkazem:

```
hciconfig hci0 up
hciconfig hci1 up
```

Tuto činnost musíme provádět po každém spuštění počítače a také při každém připojení i odpojení zařízení ze zdířky. Může nastat i případ, kdy systém napíše, že je zařízení ve stavu UP, ale nezačne reagovat na příkazy, dokud mu nepošleme příkaz:

```
hciconfig hci1 down
```

A následně příkaz UP. Tím dojde k jeho vypnutí a zapnutí a zařízení začne provádět naše příkazy.

V případě připojení více modulů je třeba Ubertooth moduly mezi sebou identifikovat, protože ve všech případech se systému hlásily hromadně pod zařízením hci0. K tomu slouží příkaz:

```
ubertooth-util -I -U0
```

Tím se rozblikají LED diody na prvním Ubertooth modulu. Pokud zvolíme parametr –U1, tak tím identifikujeme druhé zařízení a s –U2 zařízení třetí. Po restartu počítače se ale indexace může změnit, proto na to musíme pamatovat například

v momentě, kdy budeme některé zařízení používat se směrovou anténou a jiné s všesměrovou, aby tak nedošlo k mylné interpretaci naměřených hodnot.

Parametr –U používáme, pokud je v systému připojeno více Ubertooth modulů současně. V tom případě ho musíme používat u všech příkazů. Např.

```
ubertooth-dfu -U1 -d bluetooth_rxtx.dfu
```

Výše zmíněný příkaz nám určuje, že update firmware bude proveden na zařízení s indexem 1.

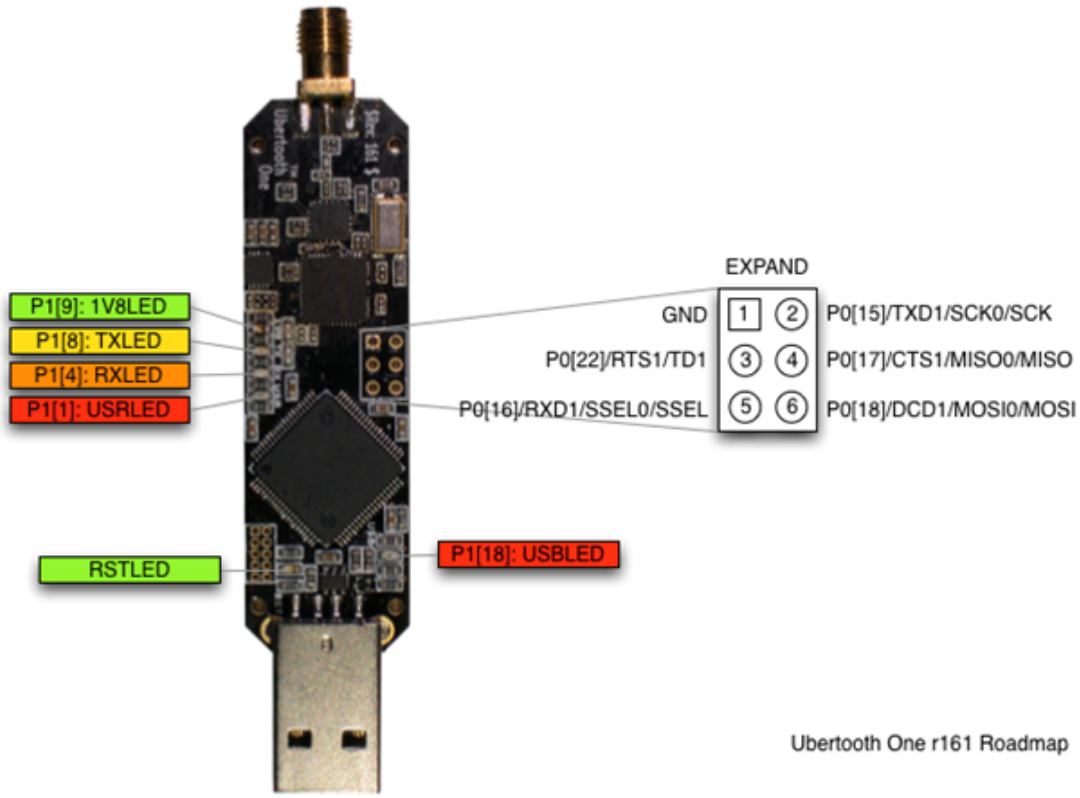
Před započetím testování je ještě vhodné pročíst si manuálové stránky všech Ubertooth nástrojů, pomocí příkazu „man“. Jsou stručné a rychle se tak zorientujeme v tom, co má který nástroj na starosti.

V poslední řadě je vhodné ještě nainstalovat poslední verzi programu „crackle“ který nám poslouží v konečné fázi 2. testu.

Instalační balík je možné si stáhnout ze stránek projektu:

https://github.com/mikeryan/crackle

Následně ho stačí rozbalit a zkompilovat stejným způsobem jako nástroje ubertooth. Poté je aplikace už dostupná přímo z konzole pod příkazem „crackle“.



Obr. 3.1: Popis jednotlivých LED diod na Ubertooth-One

3.2 Nalezení zařízení a zjištění jeho komunikačních možností.

3.2.1 Nalezení zařízení vysílající discovery rámce.

Pro jejich zjištění můžeme využít funkce programu hcitool. Pro skenování zařízení, poskytujících klasické spojení zadáme:

```
hcitool scan
```

Na výstupu dostaneme podobný výstup:

```
14:BB:6E:XX:XX:XX [TV] UE40JU6472
84:A4:66:XX:XX:XX [TV]Samsung LED4800000
```

V mém případě jsem nalezl dvě zařízení a jejich BDADDR. Jedno je televize s podporou Bluetooth připojení nacházející se ve stejné místnosti a druhé je televize v bytu o patro výše. Pokud nedostaneme žádná data a víme, že nějaké zařízení vysílá v blízkém okolí a má viditelnost nastavenou na stav: visible, je třeba si znovu zkontrolovat zda, je modul ve stavu UP.

V dalším kroku můžeme využít „inquiry scan“, který nám zjistí parametr „class“ a můžeme se tak dozvědět, o jaký typ zařízení by se mělo konkrétně jednat, popřípadě jakou nabízí službu.

```
root@kali:~# hcitool inq
Inquiring ...
  DC:EE:06:FE:D1:3A clock offset: 0x178b  class: 0x5a020c
  14:BB:6E:AC:5F:F3 clock offset: 0x61aa  class: 0x08043c
  84:A4:66:88:71:20 clock offset: 0x2ea6  class: 0x08043c
```

Při dalším skenování jsem navíc nalezl další zařízení. Podle CoD tabulky si ověřil kód třídy. Na internetu je možné najít zpracované tabulky pro rychlé vyhledání typu zařízení. Po ověření jsem určil, že třída 0x08043c by měla příslušet Samsung televizím a třída 0x5a020c by měla příslušet chytrému mobilnímu telefonu. CoD pole nejde jednoduchým způsobem změnit a ve většině případů by se muselo zasáhnout do firmwaru daného zařízení. Proto nám toto pole může pomoci, pokud nalezneme zařízení, která nejsou továrně pojmenovaná a uživatelé mají nastavené svoje názvy. U některých specializovaných zařízení může být CoD pole nastaveno na 0x000000 a tedy neobsahuje žádný bajt, který by nám dal bližší informace o zařízení.

Praktická a obsáhlá tabulka CoD i s konkrétními zařízeními je dostupná na adrese:

http://domoticx.com/bluetooth-class-of-device-lijst-cod/

Tímto způsobem můžeme nalézt zařízení se standardem Bluetooth Classic.

V druhém kroku zjistíme, která zařízení vysílají broadcast zprávy standardu Bluetooth Low Energy. K tomu znovu použijeme nástroj hcitool, ale s parametrem pro skenování Low Energy:

```
hcitool lescan
```

Po zadání příkazu nám začnou naskakovat BDADDR všech rámců které modul přijme. Pokud je v dosahu velké množství zařízení, nebo některá vysílají zprávy ve velmi krátkém časovém rozestupu, je vhodnější výstup přesměrovat do souboru.

```
hcictool lescan > LESCAN.txt
```

Obsah souboru po 4 sekundách skenování.

```
LE Scan ...
14:BB:6E:AC:5F:F3 (unknown)
14:BB:6E:AC:5F:F3 (unknown)
14:BB:6E:AC:5F:F3 (unknown)
14:BB:6E:AC:5F:F3 (unknown)
```

Podle adresy nyní víme, že jedna z televizí podporuje i Bluetooth Low Energy standard. Pole (unknown) nám značí, že se nepodařilo získat název zařízení.

Při skenování Low Energy zařízení se i po čisté instalaci systému můžeme setkat s chybovým hlášením:

```
Set scan parameters failed: Input/output error.
```

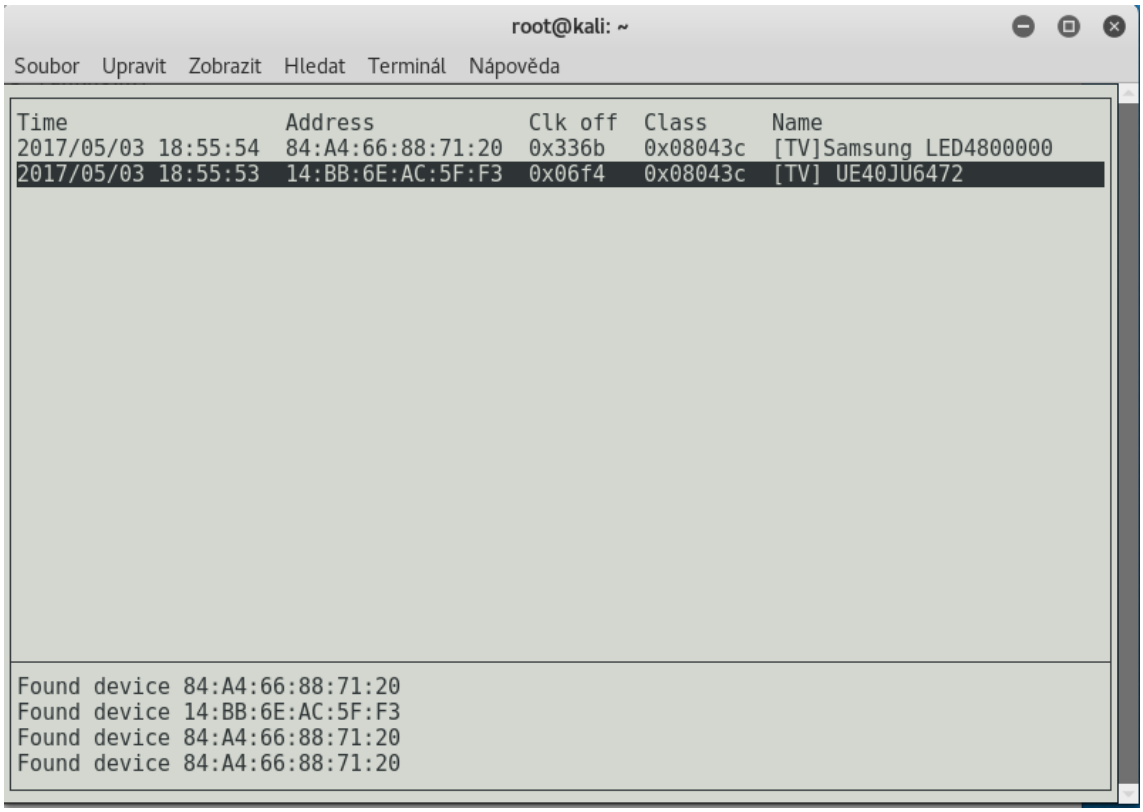
Tento problém se mi podařilo vyřešit až instalací poslední verze balíku „bluez“ z webových stránek projektu. Pro testování jsem tedy používal verzi 5.43.

Pro získání více informací o službách, které zařízení nabízí, můžeme použít aplikaci sdptool, která nám podle databáze přeloží, které funkce zařízení nabízí a jakým způsobem může komunikovat.

Lepší možností pro poloautomatické získání údajů o zařízeních je aplikace Btscanner. Spustíme ji příkazem shodným s jejím název. Po načtení databáze zvolíme stisknutím klávesy „i“ možnost „inquiry scan“. Aplikace zachytává všechny přijaté discovery zprávy na protokolu Bluetooth Classic a zároveň porovnává údaje se svojí databází (Obr. 3.2). Potom po zvolení některého z nalezených zařízení vypíše získané údaje o službách a možnostech spojení, která zařízení nabízí. Zároveň dokáže u zařízení zjistit, zda neobsahují některou ze známých zranitelností starších verzí Bluetooth a její přímé zneužití. To může být velmi užitečné zvláště při testování starších mobilních telefonů.

Aplikace může být vhodná pro rychlé zachycení velkého množství údajů v krátkém časovém úseku. Podporuje uložení údajů a tím i jejich pozdější analýzu. To se může hodit na veřejných prostranstvích, kde nemáme tolik času pro zadávání a analýzu sekvence příkazů pro jednotlivá zařízení.

Nevýhodou aplikace Btscanner je problém vytěžování procesoru, kdy po delším skenování a sbírání informací z nejasného důvodu i po ukončení zůstane běžet na pozadí a v mém případě vytěžoval cca 60% procesorových prostředků. Proto je dobré si kontrolovat, zda po jeho ukončení nezůstal stále běžet na pozadí.



Obr. 3.2: Program Btscaner

Kombinací těchto způsobů můžeme získat maximum informací o zařízeních, která data o sobě vysílají v určitých periodických intervalech. Všechny dosud zmíněné operace lze provádět obyčejným Bluetooth modulem.

3.2.2 Nalezení skrytého zařízení

V praxi se častěji setkáme s potřebou zjistit informace o zařízení, které je o sobě nevysílá pomocí broadcastových zpráv. Abychom tohoto docílili, můžeme k našemu účelu použít právě zařízení Ubertooth které nám to umožňuje.

Za tímto účelem použijeme aplikaci „ubertooth-scan“. Pro první skenování použijeme následující syntax příkazu:

```
ubertoot-scan -x -t 30 -U1
```

Tímto příkazem spustíme skenování zařízení, která již nějakým způsobem komunikují, ale nejsou objevitelná pomocí klasického discovery procesu. Tuto funkci zajišťuje parametr „-x“. Pomocí parametru „-t“ nastavujeme dobu trvání skenování na 30 sekund. A parametr „-U1“ nám udává, který připojený Ubertooth modul má operaci vykonat. Pokud v době skenování probíhá nějaká komunikace, dostaneme výstup podobný tomuto zkrácenému:

```
root@kali:~# ubertooth-scan -x -t 60 -U1

Ubertooth scan
systime=1495231138 ch=70 LAP=b2665a err=0 clk100ns=3868444
0 clk1=6189 s=-60 n=-55 snr=-5
systime=1495231144 ch=21 LAP=b2665a err=2 clk100ns=1003077
84 clk1=16049 s=-64 n=-55 snr=-9
.
.
.
systime=1495231189 ch=72 LAP=b2665a err=0 clk100ns=5461133
15 clk1=87378 s=-64 n=-55 snr=-9
systime=1495231189 ch=76 LAP=b2665a err=0 clk100ns=5503339
24 clk1=88053 s=-63 n=-55 snr=-8
UAP = 0xf7 found after 21 total packets.

Scan results:
??:??:F7:B2:66:5A [unknown]
Requesting information ...
  BD Address:  00:00:F7:B2:66:5A
  Features: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

Reading clock offset failed: Input/output error
```

Pro získání tohoto výpisu jsem spustil přesun (přibližně 100Mb) velkého textového souboru mezi dvěma chytrými mobilními telefony. Soubor je vygenerovaný pomocí programu „crunch“ a tvoří ho všechny možné kombinace písmen o délce devíti znaků, vytvořené ze slova „nemocnice“. Každé vzniklé slovo je vždy umístěno na nový řádek. Tento soubor nám bude později sloužit i jako případný objekt pro dešifrování přenášených dat.

V průběhu skenování se Ubertooth přelaďuje na jednotlivé kanály a naslouchá komunikaci na nich. Pokud je zachycen Bluetooth rámec, tak se zaznamená systémový čas zachycení, LAP master komunikujícího zařízení, časování a úroveň signálu

a šumu přijatého rámce.

Nejdůležitější hodnotou je pro nás LAP. Jedná se o poslední tři bajty Bluetooth adresy. Tyto tři bajty se přenášejí v záhlaví každého rámce, a proto jej dokážeme velmi jednoduše zachytit.

Pokud je komunikace intenzivní, nebo pokud naslouchámе dostatečně dlouho, tak Ubertooth aplikace dokáže z určitého počtu zachycených rámců odvodit i následný UAP bajt. Ten je už součástí první poloviny Bluetooth adresy, která je dána výrobcem. Proto po úspěšném zachycení LAP a úspěšném odvození UAP jsme schopni pomocí UAP filtrovat seznam výrobců a zkusit odvodit, o kterého výrobce zařízení může jít.

Pro vyhledávání je vhodné použít seznam všech prefixů rozdaných organizací IEEE.

http://standards-oui.ieee.org/oui/oui.txt

Pro moje nalezené UAP = F7 jsem vyfiltroval všechny následující nejpravděpodobnější výrobce:

```
18-A6-F7   (hex)    TP-LINK TECHNOLOGIES CO.,LTD.
00-24-F7   (hex)    Cisco Systems, Inc
A0-39-F7   (hex)    LG Electronics (Mobile Communications)
C8-19-F7   (hex)    Samsung Electronics Co.,Ltd
58-2A-F7   (hex)    HUAWEI TECHNOLOGIES CO.,LTD
```

Pomocí příkazu „l2ping“ si můžeme ověřit, zda zařízení s daným UAP a LAP opravdu existuje a zda bude odpovídat na námi zaslané rámce:

```
root@kali:~# l2ping 00:00:F7:B2:66:5A

Ping: 00:00:F7:B2:66:5A from 00:1A:7D:DA:71:10 (data size
44) ...
44 bytes from 00:00:F7:B2:66:5A id 0 time 11.85ms
44 bytes from 00:00:F7:B2:66:5A id 1 time 23.93ms
44 bytes from 00:00:F7:B2:66:5A id 2 time 23.48ms
sent, 3 received, 0\% loss
```

Nyní si pomocí následujícího příkazu zkusíme zjistit jeho jméno:

```
hcitool name 00:00:F7:B2:66:5A
```

A dostaneme odpověď „G620S-L01_01“

Zadáním názvu do internetového vyhledávače získáme jako výsledek konkrétní model mobilního telefonu Huawei. V nabídce výrobců je zastoupena i firma Huawei a můžeme tedy prohlásit, že komunikujícím zařízením je velmi pravděpodobně chytrý telefon této firmy.

Pokud chceme tento proces provádět efektivněji, je vhodnější skenování po mnohem delší dobu a výstup směrovat mimo konzoli přímo do souboru.

Po ukončení skenování je pro nás mnohem jednodušší výsledky prohlížet, filtrovat v nich a analyzovat. Také se mnohem lépe dají sdílet a rychle přenášet.

Pro dlouhodobější sběr doporučuji příkaz:

```
ubertooth-scan -x -t 600 -U1 > SCAN.txt
```

Skenování a sběr bude probíhat po dobu 10 minut a máme tak velkou šanci zachytit dostatek rámců pro identifikaci jak potřebných rámců s LAP v hlavičce, tak i následné odvození UAP.

3.3 Zachycení dat u Bluetooth classic

Pro pokus o zachycení dat opět spustíme přenos souboru mezi dvěma chytrými mobilními telefony.

Pro zachytávání dat potřebujeme obě důležité části Bluetooth adresy a to jak LAP, tak i UAP. Pokud by se nám nepodařilo během skenování odvodit UAP a máme k dispozici pouze část LAP, můžeme spustit aplikaci „ubertooth-rx“ pouze s parametrem LAP a program se bude snažit zachytit časování a následně odvodit i bajt s UAP.

Příkaz spustíme následovně:

```
ubertooth-rx -l b2665a
```

Byla zadána stejná LAP adresa, kterou jsme dostali jako výstup příkazu „ubertooth-scan“. V konzoli je nyní možné pozorovat obdobný výpis, navíc obohacený o informace o časování.

```
.
.
offset < CLK_TUNE_TIME
CLK100ns Trim: 5300
Clock drifted -950 in 28.536875 s. -3 PPM too slow.
systime=1495272385 ch=68 LAP=b2665a err=0 clkn=123204
clk_offset=2245 s=-51 n=-55 snr=4
systime=1495272386 ch=46 LAP=b2665a err=0 clkn=126268
clk_offset=2235 s=-56 n=-55 snr=-1
UAP = 0xf7 found after 4 total packets.
.
.
```

Důležitý je pro nás hlavně poslední řádek, ve kterém dostáváme hodnotu UAP=f7. Abychom si ověřili, že je hodnota správná, můžeme opět použít příkaz „l2ping“ ve formátu 00:00:UAP:LAP. Pokud přijímáme odpovědi, tak je adresa správná.

Nyní jsme ověřili, že máme jak LAP, tak UAP našeho hledaného master zařízení a můžeme se pokusit o zachycení datové komunikace.

V dalším kroku přidáme k příkazu „ubertooth-rx -l b2665a“ ještě parametr „-u“, za který přiřadíme UAP a parametr „-q“, kterým zvolíme soubor, do kterého budeme data ukládat ve formátu PCAP.

Výsledný příkaz může být v následujícím formátu:

```
ubertooth-rx -u f7 -l b2665a -q RXSNIFF.pcap -U1
```

Do konzole začneme přijímat velké množství informací, důležitým momentem dlouhého výpisu je ale následující část:

```
.
.
systime=1495274474 ch=29 LAP=b2665a err=0 clkn=577718
clk_offset=2439 s=-68 n=-55 snr=-13
CLK6 = 0x3b found after 5 total packets.

Calculating complete hopping sequence.
Hopping sequence calculated.
26402 initial CLK1-27 candidates
systime=1495274483 ch=46 LAP=b2665a err=0 clkn=578774
clk_offset=2780 s=-37 n=-55 snr=18
.
.
```

Nyní víme, že z dosavadních zachycených rámců a jejich časově frekvenční posloupnosti jsme dokázali odvodit tzv. Hopping sequence, tedy sekvenci pro algoritmus frekvenčního skákání.

Bouhužel zde ale nastává hlavní potíž.

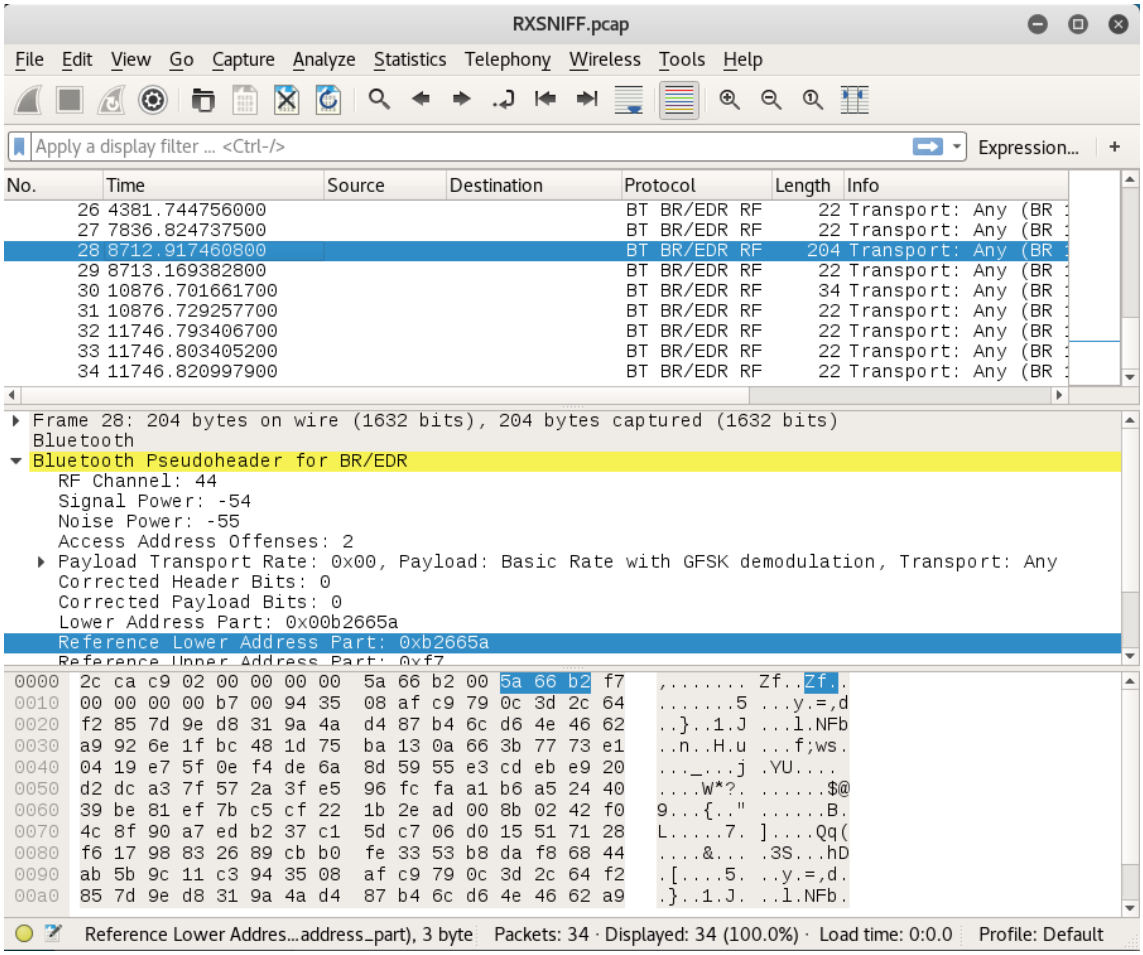
V tomto okamžiku bychom měli být schopni naslouchat na daném kanálu, na kterém bude vysílač aktuálně vysílat. Problém zřejmě nastává ve zpoždění v komunikaci mezi Ubertooth modulem a počítačem přes USB sběrnici a následné předávání do virtuálního prostředí. Nastává zde podobný jev, jako když je provozován NTP server na virtuálním počítači. U něj dochází z principu ke zpoždění předávání mezi fyzickým a virtuálním počítačem a tedy i následnému časovému rozdílu. Z toho důvodu usuzuji, že podobný efekt doprovází i zachytávání Bluetooth rámců.

Tento časový rozdíl způsobuje u naslouchajícího Ubertooth modulu posun časové základny. Tím dochází k rozdílu funkčních hodnot časově frekvenční posloupnosti vysílače a naslouchajícího Ubertooth modulu v daném čase.

Důsledkem toho i po přenesení cca 10Mb souboru máme zachycených pouze cca 30 rámců a celkovou velikost jen několik málo kilobajtů.

Z toho důvodu by bylo vhodnější provést tento test na fyzickém počítači, na němž bude spuštěn systém Kali linux a bude přímo komunikovat s hardwarem.

I tak ale máme zachycená data ve formátu PCAP, proto jej můžeme otevřít a jednodušeji analyzovat pomocí programu Wireshark.



Obr. 3.3: Analýza zachyceného pcap souboru pomocí programu Wireshark

V analyzovaném souboru jsem našel pouze 3 rámce s datovým polem. Konkrétně 1. a 28. rámec o celkové délce 204 bajtů a 30. rámec o délce 34 bajtů. Ostatní rámce s délkou 22 bajtů jsou pouze hlavičky. Viz. Obr.3.3

Za tímto účelem jsem provedl čistou instalaci a základní nastavení systému Kali Linux na svém notebooku, v němž jsem pouze vyměnil stávající pevný disk za

prázdný, určený pouze pro účely tohoto testování.

Při instalaci jsem znovu opakoval všechny instalační postupy, jak uvádím na začátku práce a ověřil jsem si, že vše bez problémů proběhlo i na jiném počítači.

Předchozí postup pro zachytávání dat jsem zopakoval za stejných podmínek. Nalezl jsem si komunikující master zařízení a spustil zachytávání jeho komunikace pomocí příkazu „ubertooth-rx“.

Přenos mezi mobilními telefony jsem opět ukončil po přenesení přibližně 10Mb dat. Výsledkem na systému instalovaném mimo virtuální prostředí jsem si potvrdil hypotézu o problému časového zpoždění.

Na začátku zachytávání si Ubertooth během několik vteřin našel časování a sekvenci pro frekvenční skákání. A následně začal přijímat a dekódovat data z linkové vrstvy a ukládat do PCAP souboru. Výpis v konzoli zobrazuje i úspěšně zachycené rámce a jejich datové pole.

Takto v konzoli vypadá výpis úspěšně zachyceného rámce.

```
systime=1495298726 ch=50 LAP=b2665a err=0 clkn=17950740
clk_offset=2337 s=-48 n=-55 snr=7
Packet decoded with clock 0x0a (rv=2)
  Type: DH5/3-DH5
  LT_ADDR: 2
  LLID: 1
  flow: 1
  payload length: 276
  Data: 85 68 e4 b2 81 eb 62 86 aa cc 46 40 9e 48 7d ea 80
  d0 bc 56 46 df cb 72 e9 78 dc 28 7d 15 da 73 6a 06 5b 17
  13 81 64 79 87 3f 6e 94 be 0a ed 39 35 83 ad 8b 89 40 b2
  bc c3 1f 37 4a 5f 85 f6 9c 9a c1 d6 c5 44 20 59 de e1 8f
  1b a5 af 42 7b 4e cd 60 eb 62 22 90 2c ef f0 c7 8d d2 57
  a1 3d a7 66 b0 75 31 11 48 96 77 f8 e3 46 e9 ab d0 9e 53
  33 d8 ba 98 08 24 cb 3b fc 71 a3 f4 55 68 cf a9 19 6c 5d
  4c 04 92 e5 1d fe b8 51 fa 2a b4 e7 d4 0c b6 2e 26 02 c9
  f2 0e 7f dc 28 7d 15 da 73 6a 06 5b 17 13 81 64 79 87 3f
  6e 94 be 0a ed 39 35 83 ad 8b 89 40 b2 bc c3 1f 37 4a 5f
  85 f6 9c 9a c1 d6 c5 44 20 59 de e1 8f 1b a5 af 42 7b 4e
  cd 60 eb 62 22 90 2c ef f0 c7 8d d2 57 a1 3d a7 66 b0 75
  31 11 48 96 77 f8 e3 46 e9 ab d0 9e 53 33 d8 ba 98 08 24
  cb 3b fc 71 a3 f4 55 68 cf a9 19 6c 5d 4c 04 92 e5 1d fe
  b8 51 fa 2a b4 e7 d4 0c b6 2e 26 02
```

Z výpisu můžeme zjistit, o jaký typ se jedná, jaká je celková délka rámce a také datové pole se zašifrovanými daty v hexadecimální podobě.

Výsledný zachycený soubor má velikost okolo 2,3Mb což je způsobeno jednak spuštěním zachytávání o něco málo později po započetí přenosu, dále časem kdy Ubertooth zjišťuje frequenci hopping sekvenci a poškozenými rámci které byly zahozeny. Skenování totiž probíhá v pasivním režimu, a tedy nijak není možné zasahovat do provozu, nebo nějakým způsobem si znovu vyžádat chybně přijaté rámce.

Posledním krokem je dešifrování párování, linkového klíče a zachycených dat.

K tomu je možné použít aplikaci například aplikaci Btcracker. Tento program nám dokáže spočítat PIN, který byl použitý v průběhu zachyceného párování. To ale platí pouze o Bluetooth verzích 1.1 a 2.0, u kterých je tento problém s párováním znám. Pro vyšší verze už tuto aplikaci použít nemůžeme.

U vyšších verzí tedy není žádný jednoduchý způsob, jak by bylo možné data dešifrovat. Můžeme se ale setkat i se zařízeními, která mají chybně implementované Bluetooth, nebo z nějakého důvodu šifrování na linkové vrstvě nepoužívají. I z tohoto důvodu je vhodné pro jistotu šifrovat na aplikační vrstvě, pokud to dovolují procesorové a energetické možnosti daného zařízení.

3.4 Zachycení párování u Bluetooth Low Energy

Pro tento účel nám bude sloužit aplikace „ubertooth-btle“. Tato aplikace má za cíl pasivní monitorování Bluetooth Low Energy komunikace.

Abychom mohli párování zachytit, musíme naslouchat na jednom, nebo ideálně na všech třech advertisement kanálech.

Pokud již známe naši cílovou adresu, tak se můžeme pustit do sestavení příkazu. V mém případě jsem zachytával na všechny tři Ubertooth moduly současně.

Nejdříve musíme určit kanál, na kterém bude daný modul naslouchat. V defaultním nastavení se jedná o kanál 37, tedy o frekvenci 2402MHz. Druhý advertisement kanál má index 38 (2426MHz) a třetí má index 39 (2480MHz). Tedy:

```
ubertoth-btle -A37
```

V dalším kroku si nastavíme mód „follow“ pomocí parametru „-f“ a cílovou adresu. Tedy:

```
ubertooth-btle -A37 -f -t(BD_ADDR) -c test.pcap -U0
```

Parametr „-t“ nám udává cílovou adresu. Ta je skrytá, protože jsem neměl možnost proceduru otestovat na jiném zařízení než na zařízení firmy Honeywell.

Rámce musíme zachytávat ve formátu PCAP (DLT_PPI) proto volíme parametr „-c“ a na konci pomocí parametru „-U0“ přidělujeme úlohu Ubertooth zařízení

které je indexováno číslem 0. Pro další moduly již měníme pouze čísla kanálů, názvy souborů a indexy Ubertooth modulů. Výsledné tři příkazy pro přehlednost spustíme ve třech samostatných terminálech:

```
ubertooth-btle -A37 -f -t(BD_ADDR) -c testA37.pcap -U0
ubertooth-btle -A38 -f -t(BD_ADDR) -c testA38.pcap -U1
ubertooth-btle -A39 -f -t(BD_ADDR) -c testA39.pcap -U2
```

Během zachytávání dostáváme do terminálu výstupy podobné tomuto:

```
systime=1495322123 freq=2426 addr=XXXXXXXXX delta_t=30.000
ms rssi=-17
00 1f 00 d0 2d 99 cb 35 02 01 02 03 03 0f 18 11 07 00 00 b
3 00 00 00 10 00 80 00 00 80 5f 9b 34 fd 70 52 a5
Advertising / AA 8e89bed6 (valid)/ 31 bytes
    Channel Index: 38
    Type:  ADV_IND
    AdvA:  XX:XX:XX:XX:XX:XX (public)
    AdvData:  02 01 02 03 03 0f 18 11 07 00 00 b3 00 00 00
    10 00 80 00 00 80 5f 9b 34 fd
        Type 01 (Flags)
           00000010
               LE General Discoverable Mode

        Type 03 (16-bit Service UUIDs)
           180f
        Type 07 (128-bit Service UUIDs)
           fd349b5f-8000-0080-0010-000000b30000

    Data:  00 d0 2d 99 cb 35 02 01 02 03 03 0f 18 11 07 00
    00 b3 00 00 00 10 00 80 00 00 80 5f 9b 34 fd
    CRC:   70 52 a5
```

Konkrétně tento výpis byl zachycený Ubertooth modulem s indexem č. 1 a naslouchajícím na kanálu č. 38. Je zde možné si také povšimnout vysoké úrovně přijatého signálu a tím si odvodit, že testování probíhalo na jednom stole.

V případě že při skenování jsou získávány hodnoty signálu velmi nízké úrovně okolo rssi=-80, je důležité si zkontrolovat, zda opravdu získáváme data z našeho cílového zařízení.

Pokud je zachytávání prováděno pouze pomocí jediného Ubertooth modulu, je velice důležité při každém započetí skenování zkontrolovat frekvenci, která se zobrazuje při přijatém rámci. Ubertooth modul může po prvním skenování zůstat stále

na stejné frekvenci a potom pokud skenujeme s parametrem „-A39“, může reálně skenovat stále kanál 37 na frekvenci (2402MHz).

Tento problém se v naprosté většině případů vyřeší pomocí resetování zařízení příkazem:

```
ubertooth-util -r
```

V případě že máme zapojeno modulů více, musíme zase určit, který modul se má resetovat. V opačném případě se setkáme s dalším chybovým hlášením.

```
ubertooth-util -r -U1
```

Během resetu se můžeme setkat s chybovým hlášením:

```
libUSB Error: Operation timed out:  (-7)
```

Nebo s chybou:

```
usb_claim_interface error -6
```

V takovémto případě nedojde ke správnému resetování a je nutné příkaz pouze zopakovat. Někdy i několikrát za sebou. Jakmile dostaneme následující výpis, tak máme zařízení úspěšně resetované.

```
root@kali:~# ubertooth-util -r -U1
Resetting ubertooth device number 1
root@kali:~#
```

V případě, že máme zaznamenán provoz na všech advertisement kanálech během párování, můžeme se pustit do pokusu o jejich dešifrování pomocí programu „crackle“.

Program crackle má jednoduchou příkazovou syntaxi. Parametrem „-i“ zadáváme vstupní PCAP soubor pomocí „-o“ výstupní.

Výsledný příkaz tedy bude vypadat následovně:

```
crackle -i test.pcap -o vystup.pcap
```

Pokud se v daném souboru nevyskytuje párovací proces, dostaneme podobný výstup:

```
root@kali:~# crackle -i test.pcap -o vystup.pcap
No connect packet found
No pairing request found
No pairing response found
Not enough confirm values found (0, need 2)
Not enough random values found (0, need 2)
No LL_ENC_REQ found
No LL_ENC_RSP found
```

```
Giving up due to 7 errors
```

Proto stejný příkaz zopakujeme pro další soubory. V případě úspěchu dostaneme výstup, ve kterém máme vypsané jak TK, tak i LTK.

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

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

Pomocí LTK tak můžeme dešifrovat již jakýkoliv další zachycený datový tok daných zařízení. To provedeme příkazem:

```
crackle -l 7f62c053f104a5bbe68b1d896a2ed49c
-i testFLOW.pcap -o vystupFLOW.pcap
```

Program „crackle“ pomocí LTK klíče dešifruje všechny platné rámce v balíku a výstup uloží. Následně můžeme data analyzovat například pomocí programu Wireshark.

3.5 Denial of Services na Bluetooth zařízení

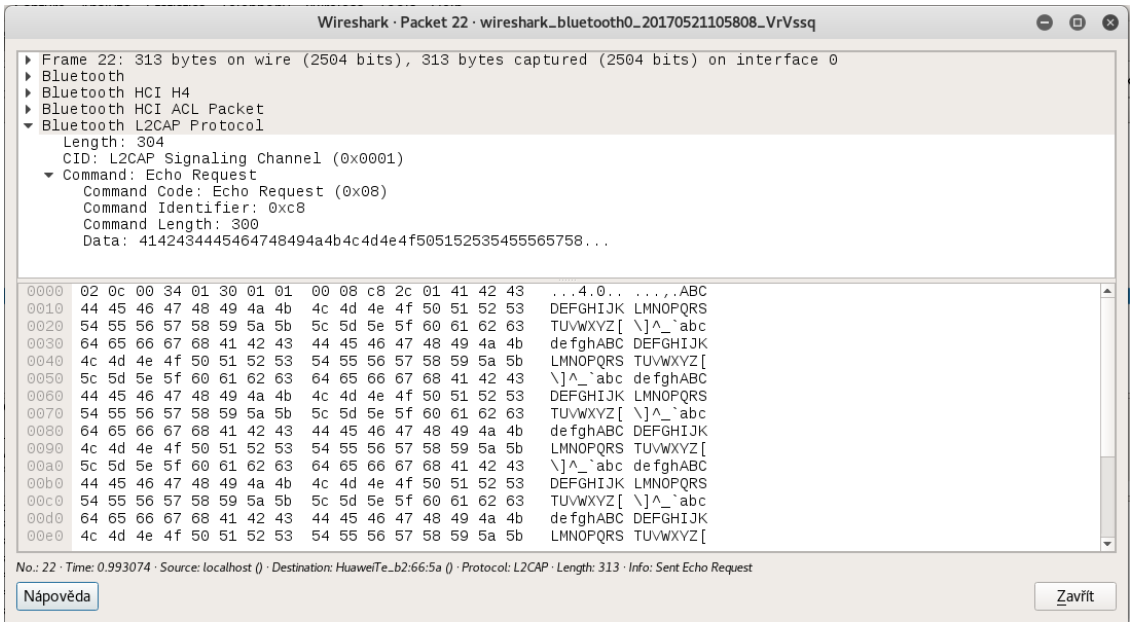
Pro tuto kapitolu testování jsem navrhl dvě metody, které mají za cíl pomocí dostupných prostředků přerušit, nebo omezit fungování a použití zařízení s Bluetooth.

3.5.1 Útok pomocí Echo request

První testovací procedura staví na zneužití hrubé síly pomocí programu „l2ping“. Tímto programem si můžeme testovat dostupnost právě Bluetooth zařízení, nad kterými neběží IP protokol a proto nemůžeme využít klasický příkaz PING, který využívá známých ICMP zpráv.

Program „l2ping“ stejně jako ICMP ping posílá echo zprávy. Ty jsou zasílány na jiná Bluetooth zařízení a čeká se na odpověď. Echo zpráva je speciální typ L2CAP rámce, jehož datové pole tvoří písmena abecedy a speciální znaky, cyklicky se opakující až do velikosti, která je požadována.

Podobu Echo zprávy zachycené a zobrazené pomocí programu Wireshark můžeme vidět na obrázku 3.4.



Obr. 3.4: Analýza zachyceného Echo Request rámce v programu Wireshark

Ze zprávy můžeme odvodit, že odeslaná datová část má velikost 300 bajtů a celková velikost zaslaného rámce se všemi hlavičkami je 313 bajtů.

Principem testu je vysílat na cílové zařízení velký počet Echo Request zpráv s co největší datovým rámcem. Tím dojde k zahlcení prostředků přijímače.

Při volbě oběti jsem zjistil, že někteří výrobci již s touto možností útoku částečně počítají a proto zařízení po přijetí například deset zpráv od jednoho zdroje spojení s ním resetují a tím dojde k přerušení odesílání.

Tuto ochranu by bylo možné jednoduše obejít pomocí skriptu, který by hlídal přerušení komunikace a v případě kdy by tato událost nastala, tak by se okamžitě pokusil o navázání nového spojení.

Úplně vypnutou funkcí pro odpovídání na Echo Request jsem zjistil u Bluetooth headsetu od firmy NGS.

Jako ideální oběť jsem si zvolil přibližně čtyři roky starý obyčejný mobilní telefon Samsung. Ten při testech spojení s útočníkem resetoval až po 30 minutách. To je dlouhá doba kdy je možné útok provádět a nebo případně spojení obnovovat i manuálně.

Po zjištění jak dlouho je možné se na zařízení dotazovat, si musíme zjistit, jakou maximální velikost můžeme na zařízení poslat. Ideální metodou jak tuto hodnotu zjistit je zvolit si počáteční velikost cca 700 bajtů a potom pomocí metody půlení intervalu zkoušet u které velikosti ještě bude zařízení odpovídat a u které již ne. Není třeba metodu aplikovat až do řádů jednotek bajtů.

U chytrého mobilního telefonu Huawei G620s jsem získal hodnotu 300 bajtů. U mobilního telefonu Samsung GTS5610 jsem se dostal až na více než dvojnásobnou hodnotu 660 bajtů.

Velkou výhodou tohoto testu je, že jde provádět prakticky s libovolným Bluetooth přijímačem a není potřeba k tomu mít pokročilé funkce které má Ubertooth, nebo jiný specializovaný hardware. Navíc lze tento test bez potíží realizovat i ve virtuálním počítači.

Výsledný příkaz pro odeslání jsem sestavil do následné podoby:

```
l2ping -i hci0 -s 660 C4:42:02:F5:CC:F2
```

Parametr „-i“ udává které Bluetooth zařízení má být použito. To využijeme v případě, kdy máme připojeno více zařízení současně.

Parametr „-s“ udává délku zprávy.

Nakonec už jen připojíme adresu cíle.

Další výhodou je, že nemusíme znát celou Bluetooth adresu zařízení. Pro potřeby odeslání stačí, znát LAP a UAP část. Takže obě hodnoty můžeme získat například pomocí metody pasivního sledování kanálu s pomocí Ubertooth a programu „ubertooth-rx“.

Příkaz tedy můžeme zadat i v podobě:

```
l2ping -i hci0 -s 660 00:00:02:F5:CC:F2
```

Požadavky tak půjdou vždy na to stejné zařízení. Prakticky je možné si změnit část NAP na libovolnou hodnotu a příkaz bude vždy fungovat. Pokud změníme některý bajt LAP, nebo UAP, tak nám příkaz pravděpodobně vypíše chybu:

```
Can't connect: Host is down
```

Schválně zmiňuji slovo pravděpodobně. Mohou nastat dva případy, kdy příkaz bude pokračovat i s touto změnou.

První možností je, že v okolí se nachází zařízení, které tuto adresu skutečně má. Obzvláště se může jednat o případ, kdy testujeme více stejných zařízení a při jejichž výrobě byly použity čipy ze stejné série. V tomto případě bychom pak mohli útok provádět na jiné zařízení, než na které chceme a dostali bychom tak zkreslené výsledky. Abychom si rychle ověřili, že se jedná o jiné zařízení, stačí provést další 2-3 pokusy s jinými změnami adresy. Pokud některé varianta bude vypisovat chybu spojení, tak jsme pouze narazili na více podobných zařízení.

Druhá možnost je chyba systému. S tímto problémem jsem se setkal pouze ve virtuálním prostředí. Chyba se projevuje tak, že i po zadání několika náhodných adres nám vždy přichází odpovědi na námi vyslané požadavky. Problém se mi podařilo vyřešit pouze pomocí restartování adaptéru a virtuálního počítače.

Pro vygenerování dostatečného množství požadavků jsem v terminálu využil možnosti řetězení příkazů pomocí operátoru „&“. Takto za sebou zadané příkazy probíhají ve stejný časový okamžik, akorát další příkazy proběhnou v dalších terminálech na pozadí, ale výstup je vypsán zpět do terminálu, ve kterém byl řetězec zadán.

Výsledný příkaz a jeho výstup může být následující:

```
root@kali:~# l2ping -s 100 58:2A:F7:B2:66:5A & l2ping -s
100 58:2A:F7:B2:66:5A & l2ping -s 100 58:2A:F7:B2:66:5A &
 l2ping -s 100 58:2A:F7:B2:66:5A & l2ping -s 100 58:2A:F7
:B2:66:5A
[1] 1344
[2] 1345
[3] 1346
[4] 1347
Ping: 58:2A:F7:B2:66:5A from 40:2C:F4:C2:C1:7D (data size
100) ...
Ping: 58:2A:F7:B2:66:5A from 40:2C:F4:C2:C1:7D (data size
100) ...
Ping: 58:2A:F7:B2:66:5A from 40:2C:F4:C2:C1:7D (data size
100) ...
Ping: 58:2A:F7:B2:66:5A from 40:2C:F4:C2:C1:7D (data size
100) ...
Ping: 58:2A:F7:B2:66:5A from 40:2C:F4:C2:C1:7D (data size
100) ...
100 bytes from 58:2A:F7:B2:66:5A id 0 time 10.21ms
100 bytes from 58:2A:F7:B2:66:5A id 0 time 10.29ms
100 bytes from 58:2A:F7:B2:66:5A id 0 time 10.15ms
100 bytes from 58:2A:F7:B2:66:5A id 0 time 10.32ms
100 bytes from 58:2A:F7:B2:66:5A id 0 time 10.10ms
```

Takto bude odesláno ve stejný čas celkem pět požadavků o velikosti 660 bajtů dat + 13 bajtů pro hlavičku každé zprávy, tedy celkem 673 bajtů * 5 = 3 365 bajtů data.

Abychom se nesnažili odeslat příliš velké množství dat v jeden časový okamžik, můžeme spustit příkaz z dalšího terminálu s určitým časovým zpožděním.

Abychom dosáhli ještě vyšší efektivity, můžeme odesílání zřetězit mezi další Bluetooth moduly. Jedna možnost je zkombinovat odesílání v jeden čas z obou zařízení:

```
l2ping -i hci1 -s 660 00:00:02:F5:CC:F2 & l2ping -i hci0
-s 660 00:00:02:F5:CC:F2
```

Druhou možností, která je lepší z důvodu větší rovnoměrnosti zatížení oběti, je odeslat jednu dávku ze zařízení „hci0“ a následně o jednu až dvě sekundy později ze zařízení „hci1“.

Pro otestování na oba dva DOS útoky jsem si zvolil výše zmíněný obyčejný mobilní telefon Samsung a Bluetooth headset NGS Arctica runner.

Pro otestování na zahlcení požadavky Echo Request jsem zvolil následující parametry. Jedna dávka pro odeslání obsahovala celkem 50 požadavků Echo Request zpráv o velikosti celkem 50 * 673 bajtů = 33 650 bajtů = 33,65 kilobajtů dat.

Na každém zařízení byla v rozestupu jedné vteřiny odeslána jedna dávka. Celkem z každého zařízení bylo spuštěno odesílání 20 paralelně běžících dávek po 50 požadavcích. Na cílové zařízení se teda odesílalo 100 požadavků/sekundu o celkové velikosti 67,30 kilobajtů/sekundu. Celková vrcholová hodnota útoku bylo 2000 požadavků o celkové velikosti 1 346 000 bajtů během dvaceti vteřin.

K mému překvapení nedocházelo k zahazování zpráv. Ovšem jako ukazatel, že útok je účinný, bylo, že přibližně od 4. odeslané dávky se začala rapidně zvyšovat odezva. V době vrcholu útoku latence činila kolem 10000 - 12000ms.

Po čase ale začalo cílové zařízení postupně resetovat příchozí spojení a útok tak po přibližně půl hodině bez dalšího lidského zásahu ustal.

Z toho důvodu jsem tento útok zopakoval a v jeho průběhu jsem se s cílovým mobilním telefon snažil navázat spojení a odeslat mu soubor o velikosti 3,14 Megabajtů.

Vyhledání zařízení a spárování během útoku proběhly na první pohled zcela v pořádku a nezaznamenal jsem žádné výrazné zdržení.

Následný přenos souboru proběhl také v pořádku. Nedošlo k přerušení spojení ani k výraznému kolísání přenosové rychlosti. Doba potřebná pro přenos souboru byla 1:49.95.

Výsledek na první pohled vypadal, že útok na komunikaci nemá prakticky žádný vliv, proto jsem provedl referenční měření se stejnými zařízeními, pouze mimo útok a o několik minut později.

Doba potřebná pro přenos souboru do zařízení, které nebylo pod útokem činila 0:22.94. Tedy útok výrazně zasáhl do přenosové rychlosti zařízení. Pokud podělíme počet vteřin při útoku a počet vteřin mimo útok, dostaneme koeficent, kolikanásobně se nám přenosová rychlost snížila. V rámci toho testu tedy došlo k 4,8 násobnému snížení přenosové rychlosti.

Na headset zařízení NGS Arctica runner se mi nepodařilo test provést, protože kvůli jednoduchosti zařízení nemá v sobě vůbec implementovanou funkci pro reakci na Echo Request požadavky.

3.5.2 Útok pomocí vytížení pásma

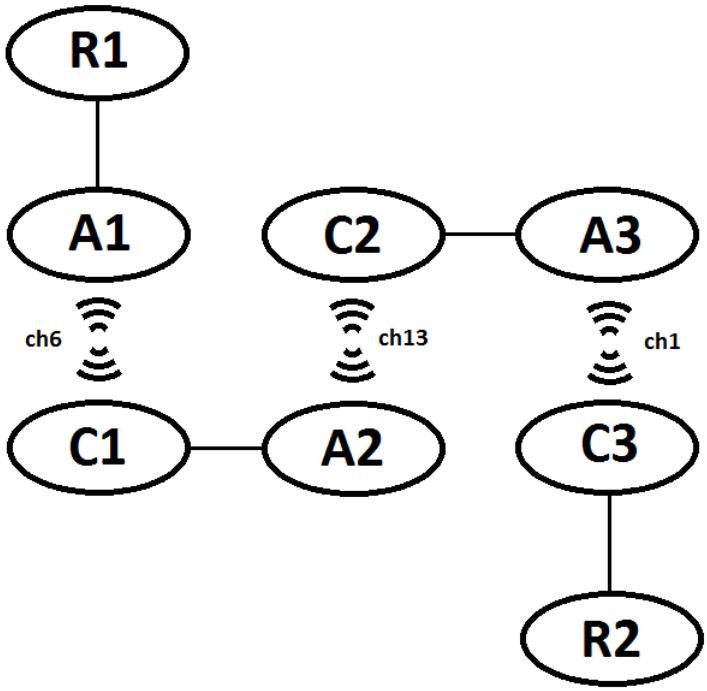
Další formou útoku je útok na vytížení pásma 2,4 GHz pomocí několika WiFi routerů zapojených do série a následný přenos velkého objemu dat skrze vytvořenou linku.

Pro tuto úlohu jsem musel použít další zařízení, která doposud nebyla nijak zmíněná a jejich použití se vyskytuje pouze v tomto testu.

Pro tyto potřeby jsem použil celkem šest WiFi routerů Netis WF2411, které jsem měl k dispozici. Hlavním důvodem jejich výběru byla cena a podpora režimu „bridge“ kdy zařízení nemá spuštěný DHCP server ani neprovádí NAT (Network Adress Translation) a na všech jeho portech včetně bezdrátového rozhraní se chová jako obyčejný switch.

Pro generování provozu napříč sítí jsem využil dva kusy Mikrotik RB 433AH.

Schématické zapojení úlohy je na obrázku 3.5.



Obr. 3.5: Schématické zapojení zařízení pro testování procedury.

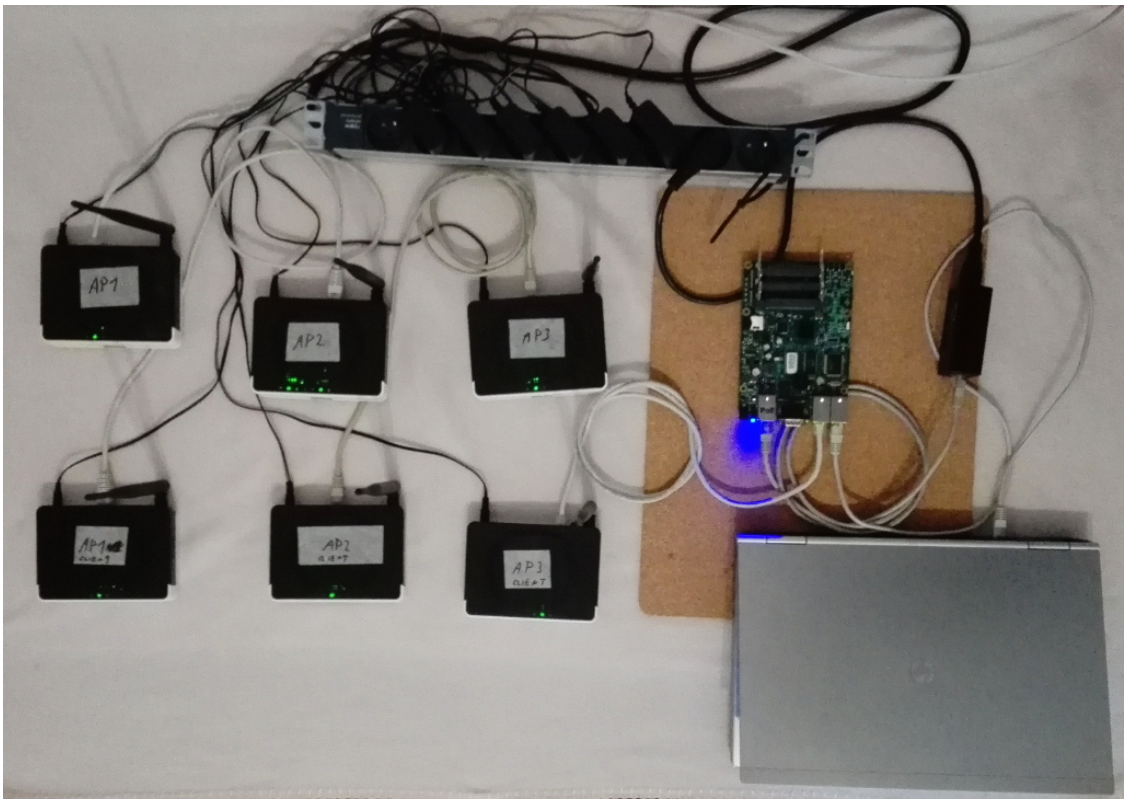
Písmeno R ve schématu označuje routery - koncové uzly, které jsou zastoupeny zařízeními Mikrotik.

Písmeno A označuje WiFi routery, jejichž bezdrátové rozhraní bylo zkonfigurováno do režimu „Access“ point a písmeno C označuje WiFi routery jejichž bezdrátové rozhraní bylo zkonfigurováno do režimu klient.

Černé propojovací čáry znázorňují kabelové spojení.

Bezdrátové propoje byly nastaveny na kanály 1, 6 a 13 s šířkou pásma 40MHz a maximálním výstupním výkon. Kanály byly zvoleny tak, aby frekvenčně bylo vykryté celé možné spektrum.

Praktické zapojení úlohy je zachycené na obrázku 3.6.



Obr. 3.6: Praktické zapojení zařízení pro testování procedury.

Router R1 byl během testu zapojen v druhé místnosti a notebook sloužil pro ovládání testu.

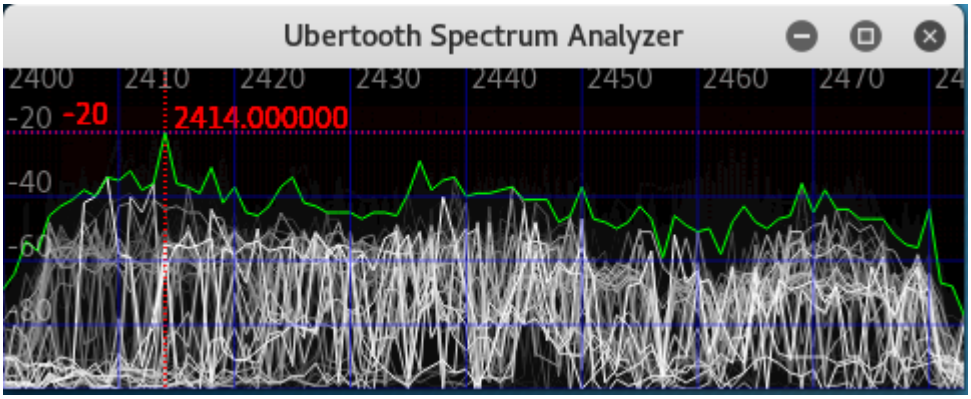
Datový provoz byl generován mezi zařízeními Mikrotik pomocí funkce „BTest“. Pro účely testu se mi osvědčilo použití základního nastavení a TCP datového toku.

V praxi je možné použít na místo mnou vybraných WiFi routerů libovolné jiné zařízení umožňující stejné funkce a místo zařízení Mikrotik mohou být například počítače, které budou zatěžovat síť pomocí programu „jperf“.

Protože jsem měl k dispozici pouze omezený počet routerů na vytvoření pouze tří bezdrátových spojení, zvolil jsem variantu tří 40MHz kanálů. Druhou variantou by bylo vytvoření celkem čtyř spojení s šířkou pásma 20MHz. A třetí nejideálnější možností, kdy by docházelo k menšímu vzájemnému ovlivnění ze strany jiných sítí, by bylo jednu z předchozích možností zprovoznit v paralelním zapojení. Tedy 3-4 paralelní datové streamy. Tak bude každý stream využívat maximální dostupnou šířku

pásma. V mém případě je linka limitována nejslabším přenosovým kanálem, který je nejsilněji ovlivněn okolními sítěmi, proto spektrum nemusí být ideálně vytížené.

Po spuštění datového přenosu napříč testovací sítí jsem si pro kontrolu udělal obrázek spektra pomocí zařízení Ubertooth, abych si ověřil, že rozdělení kanálů je správně nakonfigurované. (Obr. 3.7)



Obr. 3.7: Zobrazení spektra 2,4GHz při nadměrném vytížení.

Výsledné spektrum by mělo být vytíženo rovnoměrně v celé jeho šířce od 2400MHz až do 2480MHz. Známkou vytížení spektra na vysoké úrovni bylo, že se mi všechna zařízení odpojili od domácí WiFi sítě a v průběhu testu se ani zpět nepřipojily.

Prvně jsem znovu otestoval dobu přenosu stejného souboru o velikosti 3,14Mb do cílového mobilu.

Pokud byly oba telefony stranou testovací sítě, doba přenosu činila 1:00.21.

Druhý přenos opět stejného souboru jsem provedl s odlišným rozmístěním telefonů. Každý byl umístěn tak aby testovací síť ležela na jejich spojnici. Doba druhého přenosu tak činila 1:36.76.

Pro jistotu jsem udělal referenční přenos, kdy síť s routery byly v provozu a vysílaly, ale nebyly skrze ně přesouvány žádná data. Doba referenčního přenosu byla 00:31.69.

Z těchto poznatků si vypočítal, že útok formou vytížení pásma snižuje přenosovou rychlost přibližně 2-3 násobně.

Druhým testovaným objektem byl Bluetooth headset NGS Arctica runner.

Pro účely testování jsem nahrál zvukovou stopu s větou:

„Vytížení pásma 2,4Ghz má za následek drastické omezení propustnosti technologie Bluetooth.“

Tato zvuková stopa byla ve smyčce posílána do sluchátek.

V případě kdy byla stopa přehrávána mimo útok, nedocházelo k žádným nežádoucím jevům (praskání a vynechávání) při jejím poslechu.

Po spuštění vytížení pásma nebylo prakticky vůbec rozeznat, že jde o mluvené slovo, natož jaký je význam textu.

Pro porovnání jsou v příloze na DVD umístěny jak nahraná stopa, tak i nahrávka ze sluchátek během útoku.

Hlavní výhodou útoku je univerzálnost použití, protože je možné ho použít na kterékoliv zařízení, které komunikuje v tomto pásmu a může se jednat i o jinou technologii. Největší potíže navíc způsobí aplikacím, které jsou náročné na množství přenesených dat.

Nevýhodou je jeho náročnost na množství hardware a ve výsledku velmi nízký účinek pokud přes Bluetooth přenášíme krátké ojedinělé zprávy.

Pokud to cílové zařízení umožňuje, je proto mnohem výhodnější použít útok pomocí Echo Request zpráv.

Jako alternativu k vytěžování pásma pomocí klasických WiFi routerů by bylo použít, nebo sestavit rušičku pro pásmo 2,4GHz. Zařízení fungují na principu vysoko úrovňového generátoru bílého šumu s konkrétním frekvenčním pásmem. V rámci práce jsem se jejich využitím nezabýval. Tato zařízení ani nejsou v České republice povolena a jejich používáním se uživatel přímo vystavuje pokutě nebo trestnému stíhání.

4 TESTOVACÍ ZPRÁVA ZVOLENÉHO ZAŘÍZENÍ

Po otestování námi zvoleného zařízení známe jakou metodu útoku lze úspěšně provést. Proto je nyní musíme výsledky porovnat s praktickým využitím zařízení a jaké mohou mít tyto úspěsně provedené útoky následky.

Proto doporučuji si po skončení testů znovu projít kapitolu 1.5 a zvážit všechny oblasti kde je, nebo bude nasazeno a jaká data budou pomocí technologie Bluetooth přenášena.

Jako testované zařízení bylo na počátku zadavatelem práce firmou Honeywell zvoleno jejich zařízení.

Pomocí vytvořených postupů jsem zařízení otestoval a vytvořil testovací zprávu. Tato testovací zpráva není veřejná a je pouze pro interní potřeby firmy Honeywell.

5 ZÁVĚR

Cílem práce bylo seznámit se s problematikou bezpečnosti Bluetooth standardu a navrhout vhodné bezpečností testovací postupy pro zařízení s touto technologií.

V teoretickém úvodu byly rozebrány bezpečnostní úskalí tohoto komunikačního protokolu a v druhé a třetí části práce jsem se věnoval její praktické části - sestavení Ubertooth One adaptérů a sestavení testovacích postupů.

V průběhu jsem narazil na problém se zachytáváním rámců ve virtuálním počítači. Abych se pokusil zmenšit riziko svojí chyby během konfigurace, zkoušel jsem test na zachycení Bluetooth Classic rámců i ve virtuálním prostředí VirtualBOX a na druhém fyzickém počítači. Vždy mi test fungoval pouze přímo na fyzickém počítači. Svoji domněnku s problematikou zpoždení komunikace mezi virtuálním hardware a fyzickým hardware se mi ale nepodařilo oficálně u autorů ověřit.

Výsledkem práce je pět testovacích úloh jejichž postup sestavení a průběh testování jsem popsal co nejpodrobněji, aby bylo možné pomocí této práce postupy snadno zopakovat.

V práci jsem se uceleně věnoval dané problematice, aby čitatel získal nejen veškeré potřebné informace pro zopakování pokusů, ale také aby byl shopen správně zhodnotit veškeré aspekty, které z výsledků testů plynou a na jejich základě aby byl schopný rozhodnout zda je bezpečné pro dané použití zařízení využít.

První tři úlohy jsou jsem sestavil z jednotlivých již publikovaných postupů. Většinu postupů bylo třeba upravovat, buď z důvodu použití odlišných harwarových prostředků, nebo velmi starých postupů ve kterých byly užity již nepoužívané programy, nebo jejich velmi staré verze.

Poslední dvě úlohy jsem navrhnul z důvodu dnes množících se DOS útoků v počítačových sítích ve všemožných podobách. Chtěl bych těmito úlohami ukázat jaké nebezpečí tento typ útoku představuje i pro technologii Bluetooth a zároveň doufám že se do budoucna zvětší tlak na řešení těchto hrozeb již při návrhu nových zařízení a komunikačních protokolů.

Další rozšíření práce by bylo možné sestavením nových testovacích úloh. Vhodnými úlohami by byl útok typu Men in the middle, na senzorická zařízení a ovlivnění zasílaných hodnot. Dalším možná úloha útoku by mohla být na Bluetooth headsety a jejich odposlech v reálném čase.

LITERATURA

[1] WRIGHT, J. – CACHE, J. *Hacking Exposed Wireless, Third Edition: Wireless Security Secrets & Solutions.*. McGraw-Hill Education, 3 edition, 2015 ISBN 978-0071827638

[2] Prokopec, J. *SYSTÉMY MOBILNÍCH KOMUNIKACÍ, SÍTĚ PRO MOBILNÍ DATOVÉ SLUŽBY*. 2012 ISBN 978-80-214-4498-0

[3] Bluetooth.com: *Bluetooth 5 Now Available* [online]. Dostupné z URL: <https://www.bluetooth.com/news/pressreleases/2016/12/07/bluetooth-5-now-available/>.

[4] Bluetooth.com *Bluetooth Core Specification* [online]. Dostupné z URL: <https://www.bluetooth.com/specifications/bluetooth-core-specification/>.

[5] National Institute of Standards and Technology: *Guide to Bluetooth Security* [online]. Dostupné z URL: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-121r1.pdf/>.

[6] Wikipedie: *Bluetooth* [online]. Dostupné z URL: <https://cs.wikipedia.org/wiki/Bluetooth/>.

[7] Mike Ryan: *Bluetooth: With Low Energy comes Low Security* [online]. Dostupné z URL: <https://lacklustre.net/bluetooth/Ryan_Bluetooth_Low_Energy_USENIX_WOOT.pdf/>.

[8] Paul Walko, Tom Saul *Ubertooth Release 2017 03 R2* [online]. Dostupné z URL: <https://github.com/greatscottgadgets/ubertooth/wiki/Release-2017-03-R2/>.

[9] Kilgour Chris *Bluetooth Captures in PCAP* [online]. Dostupné z URL: <https://github.com/greatscottgadgets/ubertooth/wiki/Bluetooth-Captures-in-PCAP/>.

[10] Ryan Mike *Bluetooth PIN and LINK-KEY Cracker* [online]. Dostupné z URL: <https://github.com/mikeryan/btcrack/>.

[11] Ryan Mike *Crack and decrypt BLE encryption* [online]. Dostupné z URL: <https://github.com/mikeryan/crackle/>.

[12] Abraham John *Understanding Bluetooth Advertising Packets* [online]. Dostupné z URL: <http://j2abro.blogspot.cz/2014/06/understanding-bluetooth-advertising.html/>.

[13] Jasek Sławomir *GATTattacking Bluetooth smart devices* [online]. Dostupné z URL: <https://www.blackhat.com/docs/us-16/materials/us-16-Jasek-GATTacking-Bluetooth-Smart-Devices-Introducing-a-New-BLE-Proxy-T pdf/>.

[14] Jasek Sławomir *GATTacker* [online]. Dostupné z URL: <https://github.com/securing/gattacker//>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

TS	Time Slot - označení časového slotu
TDD	Time Division Duplex - Časové dělení komunikačního kanálu pro potřebu obousměrné komunikace.
ISM	Industrial, Scientific and Medical - označení frekvenčního pásma pro volné rádiové vysílání bez licenčních poplatků.
FEC	Forward Error Control - Zebezpečení dopředným opravným kódem.
ARQ	Accept ReQuest - příjem požadavku
GFSK	Gaussian Frequency Shift Keying - Gaussovo frekvenční klíčování - modulační technika
LE	Low Energy - typ Bluetooth spojení vyvinutý pro zařízení s potřebou bezdrátové komunikace a nízké energetické náročnosti.
EDR	Enhanced Data Rate - Bluetooth standard vyvinutý za účelem vyšší datové propustnosti.
IPv6	Internet Protocol verze 6 - adresní protokol pro komunikaci v počítačových sítích a internetu s délkou adresy 128 bitů.
HD	High Definition - vysoké rozlišení
PNP	Pseudo náhodná posloupnost
FHSS	Frequency Hopping Spread Spectrum - modulační technika užívající rozprostření spektra pomocí frekvenčního skákání.
MAC	Medium Access Control - řízení přístupu k médiu
SDP	Service Discovery Protokol - komunikační protokol pro výměnu dat o službách, které dané zařízení podporuje.
OBEX	Object Exchange - komunikační protokol na platformě Bluetooth pro přenos souborů.
RFCOMM	Radio Frequency Communication - sada přenosových protokolů, poskytující emulaci sériového portu
MITM	Men in the middle - označení pro útok typu mužem uprostřed

PDA	Personal Digital Assistant - kapesní počítač
PCB	Printed Circuit Board - deska plošných spojů
USB	Universal Serial Bus - velmi rozšířená komunikační sběrnice
BD_ADDR	Bluetooth adresa identifikující dané zařízení.
CoD	Class of Device - pole v hlavičce Bluetooth rámce označující typ komunikujícího zařízení
NTP	Network Time Protocol - síťový protokol pro synchronizaci času mezi jednotlivými účastníky
PIN	Personal Identification Number - osobní identifikační číslo
PCAP	Packet Capture - formát souboru pro ukládání zachycených paketů
TK	Temporary Key - dočasný klíč sloužící k ustanovení spojení mezi dvěma Bluetooth Low Energy zařízeními.
LTK	Long Term Key - klíč používaný k zabezpeční Bluetooth Low Energy rámců
IP	Internet Protocol - základní protokol pracující na síťové vrstvě ISO/OSI modelu.
ICMP	Internet Control Message Protocol - protokol řídících zpráv Internetu
L2CAP	Logical Link Control and Adaptation Protocol - protokol užívaný uvnitř Bluetooth rozhraní pro předávání zpráv mezi nižšími a nadřazenými vrstvami a jejich přizpůsobení.
NAP	Non-significant Address Part - první dva bajty Bluetooth adresy
UAP	Upper Address Part - třetí bajt Bluetooth adresy
LAP	Lower adress part - poslední tři bajty Bluetooth adresy
DOS	Denial of Services - forma útoku která má za cíl omezení nebo odepření služeb daného zařízení.
DHCP	Dynamic Host Configuration Protokol - protokol pro dynamickou konfiguraci zařízení v síti.
NAT	Network Adress Translation - funkce pro překladání síťových adres

TCP	Transmission Control Protocol - spojově orientovaný přenosový protokol na 4. vrstvě ISO/OSI modelu.
WiFi	Označení pro bezdrátové síťové standardy podle normy 802.11.
DVD	Digital Versatile Disk - datový nosič

SEZNAM PŘÍLOH

A OBSAH PŘILOŽENÉHO DVD

Na DVD je uložena kopie práce v elektronické podobě, soubory používání při přenosech mezi zařízeními, soubory s příkladem zachycených dat v průběhu testování úloh a audio záznamy z posledního testu - referenční vzorek a vzorek ze zařízení pod útokem.

- `/` ..... kořenový adresář přiloženého DVD
  - `headset_test` ..... audio soubory z testování headset zařízení
    - `headset_reference.m4a`
    - `headset_under_attack.m4a`
  - `sniff` ..... soubory se zachycenými Bluetooth daty
    - `BT_CLASSIC_RXSNIFF_nonvirtual.pcap`
    - `l2ping_echo_request_reply.pcapng`
  - `transfer_files` ..... soubory užívané při přenosech mezi zařízeními
    - `lorem_for_dos.txt`
    - `transfer_dict.txt`
  - `diplomova_prace.pdf` ..... elektronická kopie diplomové práce